Capturing the Geometry of Object Categories from Video Supervision

David Novotny, Diane Larlus, and Andrea Vedaldi

Abstract—We propose an unsupervised method to learn the 3D geometry of object categories by looking around them. Differently from traditional approaches, this method does not require CAD models or manual supervision. Instead, using only video sequences showing object instances from a moving viewpoint, the method learns a deep neural network that can predict several aspects of the 3D geometry of such objects from single images. The network has three components. The first is a Siamese viewpoint factorization network that robustly aligns the input videos and learns to predict the absolute viewpoint of the object from a single image. The second is a depth estimation network that performs monocular depth prediction. The third is a shape completion network that predicts the full 3D shape of the object from the output of the monocular depth prediction module. While the three modules solve very different task, we show that they all benefit significantly from allowing networks to perform probabilistic predictions. This results in a self-assessment mechanism which is crucial for obtaining high quality predictions. Our network achieves state-of-the-art results on viewpoint prediction, depth estimation, and 3D point cloud estimation on public benchmarks.

Index Terms—monocular pose estimation, monocular depth estimation, point-cloud estimation, geometry reconstruction

1 Introduction

A remarkable ability of human vision is to reliably estimate the 3D geometry of the visible objects, even from single images. Reproducing this capability in artificial vision systems has important and varied applications, such as helping robots to interact with their surroundings, driving autonomous cars through complex environments, or automatically lifting 2D movies to three dimensions.

Nowadays, mature techniques such as structure-from-motion (SfM) [1] and stereo vision [2] allow to reliably reconstruct the geometry of a *particular* scene given *several images* of it seen under sufficiently different viewpoints. Such images may be extracted as the frames of a video sequence captured by a moving camera, or collected from multiple, independent cameras looking at the same scene, famously including the example of unconstrained photos captured by tourists [3]. These reconstruction algorithms are sufficiently mature to be used in industrial applications. However, the human visual system is arguably capable of solving a significantly more complex reconstruction problem than these, namely estimating the geometry of a scene from a *single* image of it. While recovering geometry from multiple views is a matter of exploiting well defined geometric properties of the optical system formed by two or more cameras, the single-view case is inherently ill-posed. A single image is in fact insufficient to uniquely infer the shape of the objects contained in it (e.g. it is not possible to distinguish between an image of a 3D scene or the image of a photo of it, which is is flat). However, statistical reconstructions are still possible provided that one can exploit the regularity of the geometric patterns that exist in the visual world.

An important source of regularity in 3D reconstruction, and image understanding in general, is the existence of *object categories*. The reason is that objects of the same category usually have similar 3D shapes and share a common object-centric coordinate frame. Thus, identifying an object in an image provides a strong constraint in the reconstruction process, significantly reducing uncertainty. However, doing so requires to model and learn the distribution of possible 3D shapes of the objects of a given category, which is a significant challenge in its own right.

Most approaches to learning 3D categories make use of high quality but expensive supervision. CAD models have been used to fully supervise models to recognize the object viewpoint and 3D shape from a single image [4], [5]. Alternatively, standard image datasets such as PASCAL VOC [6], augmented with additional annotations such as object segmentations and keypoints [7], have been used as a supervisory signal. Whether synthetically generated or manually collected, these annotations have helped to overcome the significant challenges of learning 3D object categories, by making available to the learner ground-truth information about viewpoint, geometry, or both.

In this paper, we aim at *significantly lowering the level of supervision* required to learn the 3D geometry of object categories. In particular, we propose an *unsupervised method* (Fig. 1) that replaces synthetic or manual supervision with *motion*. Humans understand visual scenes by experiencing them from different angles, as these diverse viewpoints provide very strong cues on the geometry of specific object instances. They can then generalize such cues to properties of object categories in general. Our goal is to mimic such interaction and learn the 3D geometry of object categories using videos and no manual annotations.

- *D. Novotny is with the VGG, University of Oxford, UK, and NAVER LABS Europe, Meylan, France. E-mail: david@robots.ox.ac.uk*
- *D. Larlus is with NAVER LABS Europe, Meylan, France. E-mail: diane.larlus@naverlabs.com*
- *A. Vedaldi is with the VGG, University of Oxford, UK. E-mail: vedaldi@robots.ox.ac.uk*

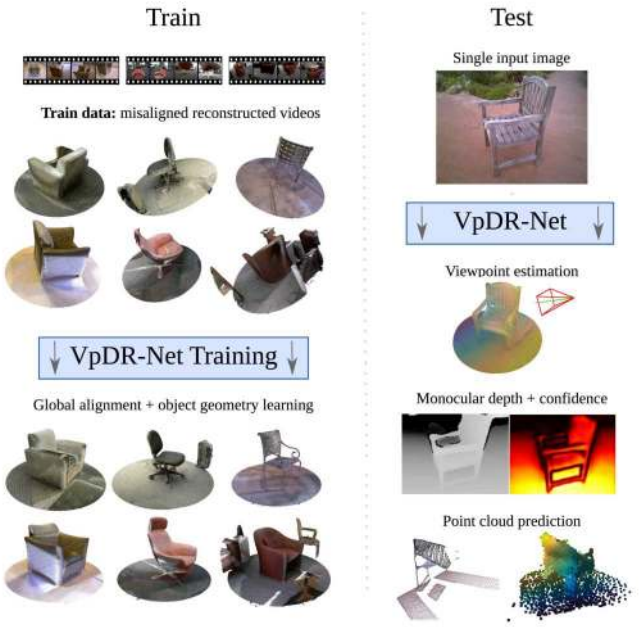

Fig. 1. We propose an architecture to learn the 3D geometry of object categories *from videos only*, without manual annotations. Once learned, the network can predict i) viewpoint, ii) depth, and iii) a point cloud, all from a single image of a new object instance.

In order to automatically generate a supervisory signal from videos, we apply SfM to individual video sequences obtained from moving cameras looking at various instances of a certain object category. As mentioned above, SfM performs well for reconstructing individual object instances, but it is insufficient to learn the shape of such objects *in general*. Thus, the key challenge is to integrate sequence-specific 3D reconstructions into a global geometric model of the object category. This has to be done in a sufficiently robust manner due to the significant level of noise in the SfM reconstructions. To overcome this challenge, we propose three key innovations.

The first innovation is **viewpoint factorization**, a new method to automatically learn to align video sequences of different object instances. Existing approaches to viewpoint alignment [8], [9] try to match 3D shapes by matching corresponding 3D features. We propose instead to learn a network that estimates, given a single image at a time, the *absolute* viewpoint of the object in the image. The network is trained in a Siamese configuration so that the *relative* motion between two images, which can be estimated using SfM, is reconstructed by composition of the absolute viewpoints estimated by the network. In other words, the relative viewpoint is factorized by the network in the product of two absolute viewpoints. We show that this training mechanism implicitly and globally aligns different objects instances while being simpler and more robust than alternatives.

The second innovation is an architecture that can **generate a complete point cloud** for a given object from only 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 the 3D space, akin to a flexible voxelization and a corresponding space occupancy map.

The third innovation is a technique that allows neural networks to **express uncertainty**. More precisely, our networks are designed to automatically predict probability distributions associated to their outputs, which allows them to learn from noisy annotations in a principled manner. We show that, when this mechanism is used, the networks train in a more robust manner.

All three contributions are leveraged by our proposed architecture, a deep network composed of three modules (Fig. 2). The first module estimates the *absolute viewpoint* of objects. This aligns different object instances to a common reference frame where geometric relationships can be modeled more easily. The second module estimates the 3D shape of an object from a given viewpoint, producing a *depth map*. The third module *completes the depth map to a full 3D reconstruction* in a globally-aligned reference frame. Combined and trained end-to-end without manual annotations, from video sequences alone, these components constitute our **VpDR-Net** network, which can jointly estimate the **v**iew**p**oint, the **d**epth and the 3D **r**econstruction of any new object instance from a single image.

This article is an extension and archival version of our previous work [10]. It is organized as follows. Sec. 2 reviews relevant literature. Sec. 3 presents the architecture as well as the training strategy of VpDR-Net. Sec. 4 proposes a novel probabilistic framework for improving the robustness of the VpDR-Net learning process. Sec. 5 shares the learning details. Sec. 6 validates our method empirically and sec. 7 summarizes and discusses our findings.

2 Related work

Capturing the geometry of an object category comprises several subtasks, such as predicting the viewpoint, the depth, and the 3D model of a novel object instance from a single image. This section discusses relevant prior works in these areas.

2.1 Viewpoint estimation

The vast majority of viewpoint estimation methods requires full supervision. Most methods are trained with manual pose annotations [11]–[17]. Full supervision can also be obtained by levering CAD models [14], [18], [19]. In particular, [19] automatically generates viewpoints together with rendered images. This requires to have 3D object models readily available. Both types of approaches rely on an expensive source of supervision. Less supervised, the approach of [20] produces a relative camera pose estimation but takes pairs of images as input.

Only few works have trained models for viewpoint estimation with videos sequences [8], [9]. In order to do so, [8] leverages a generative graphical model to discover object parts while [9] first reconstructs a 3D model per video sequence and then aligns these models.

The task of aligning point clouds is far from trivial. Most existing methods are highly sensitive to noise (see a review in [21]) and require high quality reconstructions. More robust to noise, [22] still requires to match objects of the same shape. Sedaghat et al. [9] solve the shape alignment problem using an appropriate global description of the point

clouds together with a global search strategy based on the pairwise alignment of these point clouds. We depart from this strategy by implicitly aligning point clouds as a part of the training of our Siamese viewpoint factorization network.

2.2 Monocular depth estimation

Depth estimation has been tackled with a large variety of approaches including structure from motion, shape-from-X, or multi-view stereo. In this work, we focus on monocular (i.e. single-view) depth estimation.

Many methods have cast monocular depth estimation as a supervised learning problem, predicting the depth of each pixel using models that have been trained on large datasets annotated with pixel-level ground-truth depth [23]–[25]. Saxena et al. [26] propose a patch-based approach that estimates the 3D location and orientation of local planes to explain each patch, leveraging a dataset of laser scans for training. The predictions are then combined together using an MRF. Liu et al. [25] use a convolutional neural network to learn the weights of the terms of the random fields. Ladicky et al. [23] incorporate semantics into their model to refine the pixel depth estimation. The approach of Karsch et al. [27] retrieves whole depth images from a training set.

More recently, deep learning architectures have been successfully trained for this task. Eigen et al. [24] use a two scale deep network trained with pixel-level depth values. Some works have combined deep architectures with random fields [28] or considered different losses [29], [30].

All these approaches require high quality, pixel aligned, ground truth depth maps at training time. Recently, several works have tackled the problem of learning depth from incomplete or no supervision. Training with image stereo pairs is addressed in [31], [32]. Zhou et al. [33] further decrease the level of supervision and learn a depth and egomotion predictor from unconstrained video sequences.

In this work, we train a neural network architecture for this task using the supervision provided by the reconstructions automatically obtained with an SfM algorithm. In order to cope with the noise in the output of SfM, we devise specific training mechanism including robust probabilistic losses. Our depth predictions are then used to initialize the ensuing 3D shape completion step.

2.3 3D shape prediction

The ability of recovering 3D geometry from a single image is a long standing and challenging problem. Many class-agnostic approaches have been proposed such as shape from shading [34], [35] or from silhouette [36], [37]. Yet, knowing the category of the object to reconstruct allows to leverage useful prior information.

Methods that use a 3D model of the target object go back to the seminal works of Roberts [38] and Lowe [39]. They recently regained popularity with the availability of datasets of 3D CAD models [4], [15]. In one line of research, methods estimate the 3D shape of objects by retrieving and aligning the most similar 3D model from a CAD library [40]–[43]. Other approaches leverage these 3D models to train a network to directly predict the 3D shape of an object in a fully supervised fashion. These methods differ in the type of representation used for the predicted 3D shape. [44], [45] predict a voxel occupancy grid. [46] alleviates the high memory footprint of the voxel-based methods by processing only the voxel grid cells that are predicted to have non-zero occupancy. The approaches from [47]–[49] predict surfaces instead of voxel grids. More related to our approach, [50] learns a variational auto-encoder which outputs a point cloud approximating the surface of an object depicted in a single input image. Yang et al. [51] propose a novel deep point cloud predictor that iteratively folds an initial fixed grid of 3D points. All these methods require handmade 3D CAD models at train time.

2.4 Data-driven approaches for category specific 3D reconstructions

Structure from motion (SfM) [1], [52], [53] can produce high quality 3D reconstruction by matching features across different views of the *same* instance. Matching between *different* instances of a category is much more challenging, and SfM methods generally have difficulties handling the intra-class variations. To overcome this issue, some approaches combine SfM with manual annotations [36], [54], such as keypoints [7], [55] to estimate a rough 3D geometry of objects for unordered sets of images from the same class.

More recently, deep networks have been combined with low-level geometry cues in order to learn category specific shape predictors. Rezende et al. [56] learn 3D structures from various levels of supervision, where the lowest level comprises multiple views of an object. Similarly, [57] exploits multi-view segmentation masks and depth maps while [58], [59] use object silhouettes. All these works assume knowledge of the ground truth camera viewpoint. In this work we do not need any additional annotations as we leverage motion cues.

3 Proposed Architecture

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 with no supervision (Fig. 2). The key insight is that, while videos do not solve the problem of relating the 3D shape of different object instances, they at least provide powerful if noisy cues about the 3D shape of the individual instances.

At training time, 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. VpDR-Net, illustrated in Fig. 2, 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. 3.1.2); ii) a *monocular depth* predictor $\Phi_{\text{depth}}(f_i^t)$ (sec. 3.2) and iii) 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. 3.3). Learning starts by preprocessing videos to extract instance-specific egomotion and shape information (sec. 3.1.1).

At test time, VpDR-Net takes a single image as input and can estimate simultaneously the viewpoint, the depth map, and the 3D reconstruction of the object contained in it.

3.1 Viewpoint prediction module

3.1.1 Preprocessing

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) the camera 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 $T_t^i \in \mathbb{R}^3$. We use the convention that g_t^i transforms world-relative coordinates $p_{\text{world}} \in \mathbb{R}^3$ to camera-relative coordinates, i.e. $p_{\text{camera}} = g_t^i p_{\text{world}}$.

We extract this information using off-the-shelf methods: the structure-from-motion (SfM) algorithm COLMAP for RGB sequences [60], [61], and an open-source implementation [62] of ORB-slam2 (OS) [63] for RGBD sequences. The information extracted from RGB or RGBD data is qualitatively similar, except that the scale of SfM reconstructions is arbitrary.

3.1.2 Intra-sequence alignment

Methods such as SfM or OS 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 [9] 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 [9].

One of our main contributions, described below, is to reverse this process by learning a viewpoint predictor *without* explicitly matching 3D shapes. Empirically (sec. 6), 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. 3.1.1 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 $\hat{g}_t^i = g_t^i h^i$. Note that h^i composes to the right: it transforms the world reference frame and then moves it to the camera reference frame.

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':

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

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

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

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

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):

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

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

where $\ln$ is the principal matrix logarithm and

$$R_{t't}^i \doteq R_{t'}^i (R_t^i)^\top, \qquad \hat{R}_{t't}^i \doteq \hat{R}_{t'}^i (\hat{R}_t^i)^\top, \tag{6}$$

$$T_{t't}^i \doteq T_{t'}^i - R_{t't}^i T_t^i, \qquad \hat{T}_{t't}^i \doteq \hat{T}_{t'}^i - \hat{R}_{t't}^i \hat{T}_t^i. \tag{7}$$

Discussion. 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 $(\hat{R}_t^i, \hat{T}_t^i)$ 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. 6.

3.1.3 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 $g_t^i(\lambda^i) = (R_t^i, \lambda^i T_t^i)$. This ambiguity leaves eq. (2) unchanged, but eq. (3) becomes:

$$\hat{T}_{t'}^i - \hat{R}_{t't}^i \hat{T}_t^i = \lambda^i (T_{t'}^i - R_{t't}^i T_t^i) \quad \Rightarrow \quad \hat{T}_{t't}^i = \lambda^i T_{t't}^i.$$

During training, the ambiguity can be removed from loss (5) by dividing vectors $T_{t't}^i$ and $\hat{T}_{t't}^i$ by their Euclidean norm so λ^i is not required to learn Φ_{vp}. Yet, λ^i is important for depth prediction, so we estimate it as well. To do so, we note that, given a pair of frames (t, t') from sequence S^i, one can estimate the sequence scale as

$$\lambda_{t,t'}^i = \frac{\|T_{t'}^i - R_{t't}^i T_t^i\|}{\|\hat{T}_{t'}^i - R_{t't}^i \hat{T}_t^i\|}. \tag{8}$$

This expression allows to conveniently estimate λ^i as a moving average during the training iterations, as sample values of $\lambda_{t,t'}^i$ can be computed for free when training ϕ_{vp}. Note that $\lambda^i = 1$ for OS sequences with metric depth.

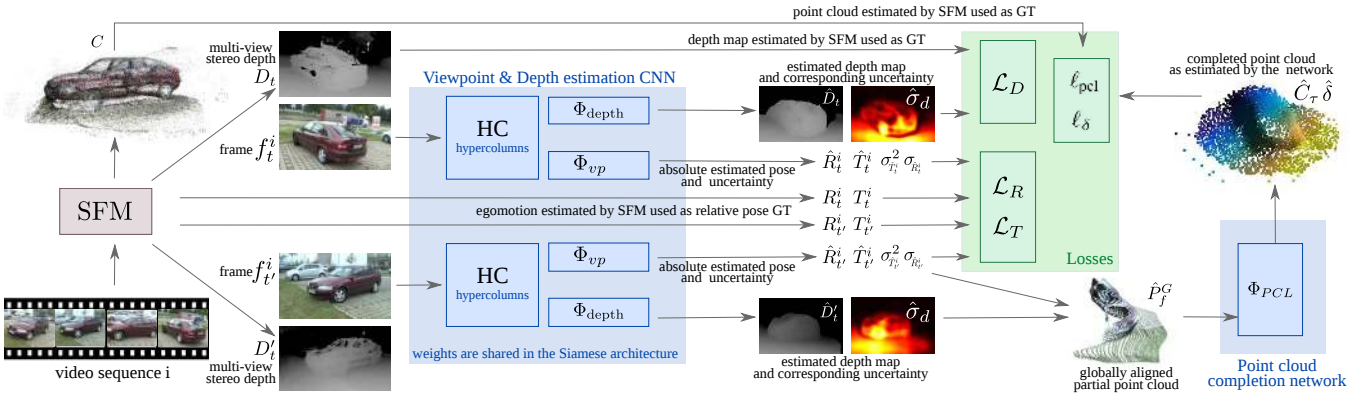

Fig. 2. **Overview of our architecture.** As a preprocessing, structure from motion (SfM) extracts egomotion and a depth map for every frame. For training, our architecture takes pairs of frames f_t, $f_{t'}$ and produces a viewpoint estimate, a depth estimate, and a 3D geometry estimate. At test time,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

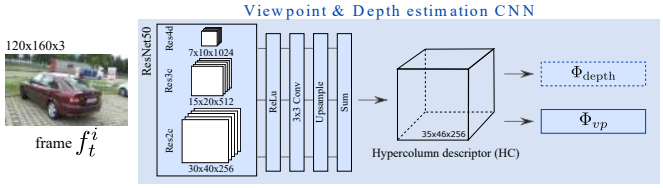

Fig. 3. **The core architecture of VpDR-Net.** This figure describes the architecture of the hypercolumn (HC) module.

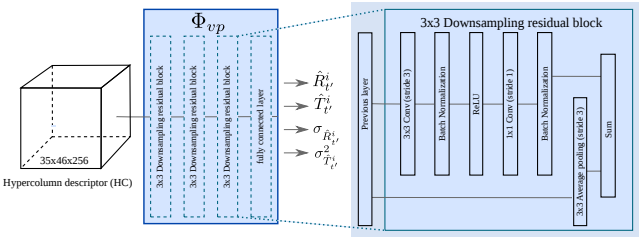

Fig. 4. **The architecture of Φ_{vp}.** Left: the layers of Φ_{vp}. Right: detail of the 3×3 downsampling residual block.

3.1.4 Architecture

The viewpoint estimation branch Φ_{vp} of our network is a convolutional architecture. Its lower part is shared between the viewpoint and the depth prediction branches. It is a variant of ResNet-50 [64] with some modifications to improve its performance as viewpoint predictor. First, in order to decrease the degree of geometrical invariance of the network, we replace all 1×1 downsampling filters with full 2×2 convolutions. We then add bilinear upsampling layers that first resize features from three different layers of the architecture (res2c, res3c, res4d) into fixed-size tensors and then sum them in order to create a multiscale intermediate image representation which resembles hypercolumns (HC) [65]. An extension of Fig. 2 that illustrates these layers responsible for the computation of the HC multiscale representation can be found in Fig. 3.

The upper part of Φ_{vp} is specific to the viewpoint prediction branch. HC is followed by 3 modified 3×3 downsampling residual layers that produce the final viewpoint prediction. While the standard downsampling residual layers do not contain the residual skip connection due to different sizes of the input and output tensors, here we retain the skip connection by performing 3×3 average pooling over the input tensor and summing the result with the result of the second 3×3 downsampling convolution branch. We further remove the ReLU after the final residual summation layer. Fig. 4 contains an overview of the viewpoint estimation module together with a detailed illustration of the modified downsampling residual blocks.

3.2 Depth prediction branch

An estimation of the viewpoint of an object is already a powerful geometrical cue allowing to relate it to a 3D scene. In this section, we describe the second branch of our network, a depth prediction module that estimates the 3D structure of the part of the object that is visible in the image.

Monocular depth prediction. The depth predictor module Φ_{depth} of VpDR-Net takes individual frames f_t^i and outputs a corresponding depth map $\hat{D}_t = \Phi_{\text{depth}}(f_t^i)$, performing monocular depth estimation. The depth map $\hat{D}_t$ is the same size as the input image and gives, for each pixel, an estimation of its distance from the camera.

In order to learn Φ_{depth} a standard approach is to minimize a distance metric between the predicted depth $\hat{D}_t$ and the ground truth D_t. Recently, [30] proposed to use the BerHu loss - a reversed version of the Huber loss which adaptively sets the cut-off threshold where the loss transitions from the ℓ_1 into the ℓ_2 part. Note that although VpDR-Net does not use this type of loss, here we describe the approach of [30] as it is later used as a non-probabilistic baseline we compare against.

Architecture. The architecture of Φ_{depth} shares the early HC layers with the viewpoint factorization network Φ_{vp}. The remainder of the pipeline is based on the state-of-the-art depth estimation method of [30]. More precisely, the network is composed of two standard residual blocks, two 2×2 up-projection layers similar to the ones from [30], leading to a 64-dimensional representation of the same size as the input image. These layers are followed by a 1×1 convolutional filter that predicts the depth map $\hat{D}_t$. This is illustrated in Fig. 5.

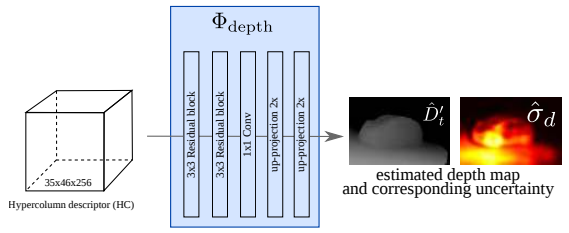

Fig. 5. **The architecture of Φ_{depth}.** Illustration of the layers specific to the depth prediction branch which predicts a depth map (sec. 3.2) and optionally an uncertainty map (sec. 4.3)

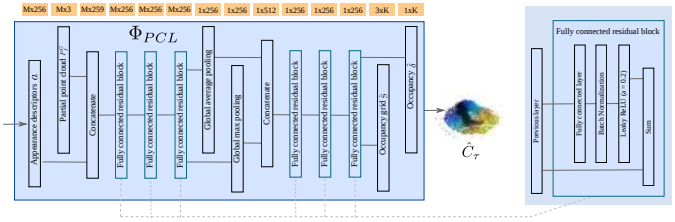

Fig. 6. **The architecture of Φ_{pcl}.** Left: The overview of the point cloud completion network, right: A detail of the fully connected residual block. Orange boxes denote the sizes of the layer outputs.

3.3 Point-cloud completion branch

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. 3.1.2 and 3.2. 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.

3.3.1 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

$$\hat{P}_f \doteq \{\hat{p}_j : j = 1, \dots, HW\}, \qquad \hat{p}_j \doteq K^{-1} \begin{bmatrix} u_j \\ v_j \\ \hat{d}_i \end{bmatrix},$$

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 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 $\hat{P}^G_f = \hat{g}^{-1}\hat{P}_f$, where $\hat{g} = \Phi_{\text{vp}}(f)$ is the camera pose estimated by the viewpoint-prediction network.

3.3.2 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 $\hat{P}^G_f$ 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:

$$\ell_{\text{pcl}}(\hat{S}) = \frac{1}{|C|} \sum_{c \in C} \min_{m=1,\dots,M} \|c - \hat{s}_m\|_2 \,. \tag{9}$$

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 $\Phi_{\text{pcl}}(\hat{P}^G_f)$ predicts additional scalar outputs

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

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^γ_m by minimizing the squared error loss $\ell_\delta(\hat{\delta}, \delta) = \sum_{m=1}^M (\hat{\delta}_m - \delta^\gamma_m)^2$. Here, raising δ_m to the power of $\gamma < 1$ prevents overweighting the support points with excessive probability masses which often correspond to "sinks" for the distant outlier points. Furthermore, δ^γ_m can be interpreted as a probability of at least one surface point being present in the vicinity of s_m. We set $\gamma = 1/3$ in our experiments.

Given the network prediction $(\hat{S}, \hat{\delta}) = \Phi_{\text{pcl}}(\hat{P}^G_f)$, the completed point cloud is then defined as the subset of points $\hat{C}$ that have sufficiently high occupancy, defined as: $\hat{C}_\tau = \{\hat{s}_m \in \hat{S} : \hat{\delta}_m \geq \tau\}$ 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 [66], 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}, the point cloud completion network Φ_{pcl} is not convolutional but uses a sequence of residual fully connected layers to process the 3D points in $\hat{P}^G_f$, 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.

In more details, the network starts by appending to each 3D point $\hat{p}_i \in \hat{P}^G_f \subset \mathbb{R}^3$ an appearance descriptor a_i and processes this input with an MLP with an intermediate pooling operator:

$$(\hat{S}, \hat{\delta}) = \Phi_{\text{pcl}}(\hat{P}^G_f) = \text{MLP}_2 \left(\operatorname*{pool}_{1 \leq i \leq |\hat{P}^G_f|} \text{MLP}_1(\hat{p}_i, a_i) \right).$$

The intermediate pooling operator, which is permutation invariant, removes the dependency on the number and order of input points $\hat{P}^G_f$. In practice, the pooling operator

uses both max pooling and sum pooling, stacking the results of the two.

For the appearance descriptors, recall that each point $\hat{p}_i$ is the back-projection of a certain pixel (u_i, v_i) in image f. To obtain the appearance descriptor a_i we reuse the HC features from the core architecture and sample a column of feature channels at location (u_i, v_i) using bilinear sampling. Note that, following [67], the fully connected residual blocks contain leaky-ReLUs with the leak factor set to 0.2. A diagram depicting Φ_{pcl} can be found in Fig. 6. The architecture is configured to predict $M = 10^4$ points $\hat{S}$.

Point cloud sub-sampling. During training the incomplete point cloud $\hat{P}_f^G$ is downsampled by randomly selecting $M = 10^4$ points based on their depth prediction confidence as estimated by Φ_{depth}. This allows the network to implicitly discard background points (as these are assigned low confidence by depth prediction). Due to this reason, at test time, the point cloud sub-sampling is also used with $M = 10^4$.

4 Probabilistic Learning

4.1 Motivation

In the previous section we have presented a basic version of our VpDR-Net network. Although such architecture can be expected to converge and subsequently perform well in standard fully-supervised settings, note that our supervisory signal can contain a significant amount of noise as it is obtained automatically by applying 3D reconstruction to RGB or RBGD images with the COLMAP [60] and ORB-slam2 [63] algorithms respectively. Typically, reconstruction methods fail for transparent regions or around specularities.

Hence, one of our key contributions, which is described in this section, consists of allowing our VpDR-Net to explicitly express this uncertainty in the ground-truth and subsequently use it in order to: (1) obtain more robust training losses; and (2) enable our model to predict the degree of reliability of the predictions.

In order to do so, we present a generic probabilistic framework where, rather than directly predicting the target values, we instruct our network to predict parameters of a distribution that approximates the predicted values. Once our regressor predicts such parameters, the actual output value corresponds to the mean of the predicted distribution (i.e. the most likely value of the distribution), while the variance of the distribution defines how concentrated is the probability mass around the most likely value, hence can be interpreted as a degree of uncertainty.

In what follows, we present a probabilistic extension of the architecture and original training losses described in the previous section.

4.2 Probabilistic predictions for viewpoint estimation

Due to intrinsic ambiguities in the images or to errors in the SfM supervision (caused for example by reflective or textureless surfaces), the viewpoint prediction branch of our network 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.

Recall that the viewpoint prediction branch Φ_{vp} predicts an absolute viewpoint $\hat{g}_t^i = \Phi_{\text{vp}}(f_t^i)$ for an input frame f_t^i, where the viewpoint is composed of a translation component $\hat{T}_t^i$ and a rotation component $\hat{R}_t^i$.

For the translation part, we modify the network to predict the absolute pose $\hat{T}_t^i$ as well as its associated confidence score $\sigma_{\hat{T}_t^i}$. We then model the relative translation as a Gaussian distribution with standard deviation $\sigma_T = \sigma_{\hat{T}_{t'}^i} + \sigma_{\hat{T}_t^i}$ and our model is now learned by minimizing the negative log-likelihood $\mathcal{L}_T$ which replaces the loss ℓ_T:

$$\mathcal{L}_T = -\ln \frac{1}{(2\pi\sigma_T^2)^{\frac{3}{2}}} \exp\left(-\frac{1}{2}\frac{\ell_T^2}{\sigma_T^2}\right). \tag{10}$$

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

$$\mathcal{L}_R = -\ln \frac{1}{C_R} \exp\left(-\frac{\ell_R}{\sigma_R}\right), \sigma_R = \sigma_{\hat{R}_{t'}^i} + \sigma_{\hat{R}_t^i}, \tag{11}$$

where $C_R = \sigma_R(1 - \exp(-\sigma_R^{-1}\pi))$ is a normalization term ensuring that the probability distribution integrates to one on the interval of attainable values of $\ell_R \in [0, \pi]$.

Note that this definition of the loss not only allows to predict the degree of uncertainty, but it also allows to increase the robustness of the training. This is because 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.

Modification of the architecture. On top of the use of different losses, the network architecture is slightly modified to predict confidence scores. The hypercolumns module remains unchanged. The upper part of Φ_{vp} is updated to predict four values T_t^i, R_t^i, $\sigma_{\hat{T}_t^i}$ and $\sigma_{\hat{R}_t^i}$, instead of two. The confidence scores $\sigma_{\hat{T}_t^i}$ and $\sigma_{\hat{R}_t^i}$ are predicted as the output of a soft ReLU units to ensure positivity.

4.3 Probabilistic predictions for depth estimation

Estimating depth from a single image is inherently ambiguous and requires comparing the image to internal priors of the object shape. Additionally, our supervisory signal is automatically generated from the SfM reconstructions, leading to annotation errors, as discussed in sec. 4.2.

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, we assume a Laplace distribution with negative log-likelihood loss

$$\mathcal{L}_D = \sum_{j=1}^{WH} -\ln \frac{\sqrt{2}}{2\hat{\sigma}_{d_j}} \exp\left(-\frac{\sqrt{2}\,|d_j - \hat{\lambda^i}^{-1}\hat{d}_j|}{\hat{\sigma}_{d_j}}\right), \tag{12}$$

where d_j is the noisy ground truth depth output by the reconstruction algorithm (COLMAP or ORB-slam2) for a given pixel j, $\hat{d}_j$ and $\hat{\sigma}_{d_j}$ are respectively the corresponding predicted depth mean and standard deviation. Due to a heavy presence of outliers in our ground truth depth data, we selected the Laplace distribution because it is a straightforward extension of the robust ℓ_1 regression loss.

Fig. 7. **Data augmentation**. Training samples generated leveraging monocular depth estimation (**ours**, top) and using depth from ORB-slam2 (baseline, bottom). Missing pixels due to missing depth in red.

An alternative approach consisting of extending the ℓ_2 loss into a Gaussian distribution (as done in sec. 4.2) was not considered because the ℓ_1 loss is known to be more robust to outliers than ℓ_2. We have not used an equivalent of the Laplacian distribution for viewpoint prediction due to the fact that its generalization to higher dimensions leads to non-trivial distributions [68].

The loss $\mathcal{L}_D$ depends on the relative scale $\hat{\lambda}^i$. For RGBD images and the ORB-slam2 algorithm $\hat{\lambda}^i = 1$. For RGB images and SfM, $\hat{\lambda}^i$ is estimated as explained in sec. 3.1.3.

Modification of the architecture. As before, the architecture of Φ_{depth} shares the early HC layers with the viewpoint factorization network Φ_{vp}. The remainder of the architecture is slightly extended with a second 1×1 convolutional filter that predict the confidence maps $\hat{\sigma}_{d_j}$ to complement the first 1×1 convolutional filter predicting the depth map $\hat{D}_t$.

5 Training the Model

The two previous sections described our network in detail, including the architecture of its three modules, respectively responsible for the prediction of an absolute viewpoint, a depth map, and a point cloud. These sections also discussed appropriate losses to train the network. In particular, sec. 4 described how to equip the network with a probabilistic introspection mechanism by training it with probabilistic losses. In this section, we describe implementation details that were found to be crucial for successfully training this network. First, we show how we perform data augmentation for these geometric prediction tasks (sec. 5.1) and we then provide technical details for reproducibility (sec. 5.2).

5.1 Geometry-aware data augmentation

As viewpoint prediction with deep networks benefits significantly from large training sets [19], 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 [17], [19], 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 [69], we propose to solve this problem by using the estimated scene geometry to *generate new realistic viewpoints* (Fig. 7). Given a sample frame together with its global pose and depth map i.e. a triplet (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) [70] to generate a new sample (f_*^i, g_*^i, D_*^i), warping both the image and the depth map, and computing the new global pose.

Sometimes the depth map D_t^i produced by the ORB-slam2 algorithm contains too many holes to yield satisfactory DIBR results (fig. 7, bottom); we found preferable to use the depth $\hat{D}_t^i = \Phi_{\text{depth}}(f_t)$ estimated by our network which is less accurate but more robust, containing almost no missing pixels (fig. 7, top).

5.2 Learning details

The VpDR-Net network 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.[1] When possible, convolutional filters were initialized with the ResNet50 weights pretrained on the ImageNet classification task. Note that, for the motorbike category where the dataset mostly contains extremely zoomed-in frames, we altered the predicted relative translations $\hat{T}^i_{t't} \doteq \hat{T}^i_{t'} - R^i_{t't}\hat{T}^i_t$ from eq. (7) to use the ground truth provided rotation $R^i_{t,t'}$ instead of the predicted $\hat{R}^i_{t,t'}$. In practice, this greatly improved the convergence speed for this category.

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.

Training minibatches were formed by first sampling a video sequence from a uniform distribution and then randomly picking an image from the sequence twice in order to obtain the final image pair. The batch size was set to 8 image pairs. In order to boost the invariance to input image noise, we blur each training image with a Gaussian filter who's variance is randomly sampled from the interval $(0, 1]$.

6 Experiments

In this section, after introducing the datasets we use in our experimental evaluation (sec. 6.1), we assess our approach on the three geometric inference tasks: viewpoint estimation in sec. 6.2, depth prediction in sec. 6.3, and point cloud prediction in sec. 6.4.

6.1 Datasets

Throughout the experimental section, we consider three datasets for training and benchmarking our network.

FreiburgCars (FrC) [9] is a set of RGB video sequences with the camera circling around 52 different models of cars. The length of videos ranges from 30 seconds to 2 minutes. Each video has been subsampled to roughly 1000 frames.

The **Large Dataset of Object Scans (LDOS)** dataset [71] contains RGBD sequences of man-made objects. We considered the bike, chair, and motorbike categories. We used 126,

1. The exact values of the loss weights are: $w(\mathcal{L}_T) = w(\mathcal{L}_R) = 0.001$, $w(\mathcal{L}_D) = 0.01$, $w(\ell_{pcl}(\hat{S})) = 1000$, $w(\ell_\delta) = 1$, where $w(\mathcal{L})$ is the weight set for the loss $\mathcal{L}$.

object class	test set	manual annot.	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	Yes	VPNet + Pascal3D	12.45	1.26	20.35	0.24	0.77	0.74
		No	VPNet + aligned LDOS	49.62	32.29	85.45	0.84	0.15	0.00
		No	**VpDR-Net (ours)**	**29.57**	**7.29**	**62.30**	**0.65**	**0.41**	**0.91**
chair	Pascal3D	Yes	VPNet + Pascal3D	21.63	4.14	39.61	0.48	0.45	0.82
		No	VPNet + aligned LDOS	55.55	41.06	90.94	0.88	0.18	0.00
		No	**VpDR-Net (ours)**	**33.70**	**14.23**	**57.61**	**0.72**	**0.35**	**0.34**
	LDOS	Yes	VPNet + Pascal3D	49.09	8.06	81.57	0.90	0.18	0.00
		No	VPNet + aligned LDOS	40.18	0.60	86.95	0.81	0.24	0.27
		No	**VpDR-Net (ours)**	**27.80**	**0.46**	**55.13**	**0.58**	**0.46**	**0.46**
motorbike	Pascal3D	Yes	VPNet + Pascal3D	21.74	2.64	34.95	0.43	0.56	0.98
		No	VPNet + aligned LDOS	140.21	58.45	128.37	0.99	0.00	0.00
		No	**VpDR-Net (ours)**	**68.67**	**11.88**	**92.09**	**1.05**	**0.08**	**0.52**
	LDOS	Yes	VPNet + Pascal3D	70.24	5.77	98.06	1.03	0.04	0.00
		No	VPNet + aligned LDOS	132.77	1.38	113.09	0.95	0.00	0.01
		No	**VpDR-Net (ours)**	**31.35**	**0.57**	**60.06**	**0.59**	**0.41**	**0.26**
bicycle	Pascal3D	Yes	VPNet + Pascal3D	23.76	3.09	46.29	0.59	0.43	0.95
		No	VPNet + aligned LDOS	114.51	37.25	124.36	1.02	0.00	0.01
		No	**VpDR-Net (ours)**	**81.84**	**24.35**	**91.27**	**1.15**	**0.00**	**0.05**
	LDOS	Yes	VPNet + Pascal3D	56.72	6.67	92.86	0.99	0.12	0.00
		No	VPNet + aligned LDOS	112.06	1.39	106.92	0.95	0.00	0.00
		No	**VpDR-Net (ours)**	**51.25**	**0.77**	**76.26**	**0.81**	**0.11**	**0.14**

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 unconstrained video sequences, is compared to VPNet-unsupervised trained on the same video sequences, aligned with the method of [9] (VPNet + aligned LDOS), and a fully-supervised VPNet (VPNet + Pascal3D). ↑ (resp. ↓) means larger (resp. lower) is better.

77, and 102 videos for the chair, motorbike and bike classes respectively. The average length of each video is 2383 frames which corresponds to 79.5 seconds.

The **Pascal3D** dataset [15] is a standard benchmark for pose estimation [17], [19]. For this dataset, we consider the four previously mentioned categories: cars, bikes, chairs and motorbikes. Following standard practice [17], [19] we only use non-truncated and non-occluded images from each category. We use the "train" set for training some of our baseline networks and for estimation of the global alignment transform $\mathcal{T}_G$ (see sec. A and 6.2 for details) and the held-out "val" set for evaluating performance of all the considered approaches.

For viewpoint estimation, Pascal3D already contains annotations. For LDOS, there are no such absolute viewpoint annotations. To generate ground truth annotations for evaluation, we manually aligned 3D reconstructions of 10 randomly-selected videos for each category and used 50 randomly-selected frames for each video as a test set.

For depth estimation, we evaluate on LDOS as it contains high quality depth maps which provide a suitable ground truth. We use the same 50 randomly selected frames from our pool of test videos, similar to the viewpoint estimation.

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, picking $3 \cdot 10^4$ points using random subsampling and farthest point sampling for FrC and LDOS respectively. The point clouds were then post-processed with 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, i.e. the 50 frames extracted from the 10 test videos of each category.

6.2 Pose estimation

First, we evaluate the VpDR-Net viewpoint predictor on the Pascal3D benchmark [15]. Unlike previous works [17], [19] 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.

Adjusting the Pascal3D annotations. 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 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 $\hat{R}$, 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 [17], [19] 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, the two sets of annotations are

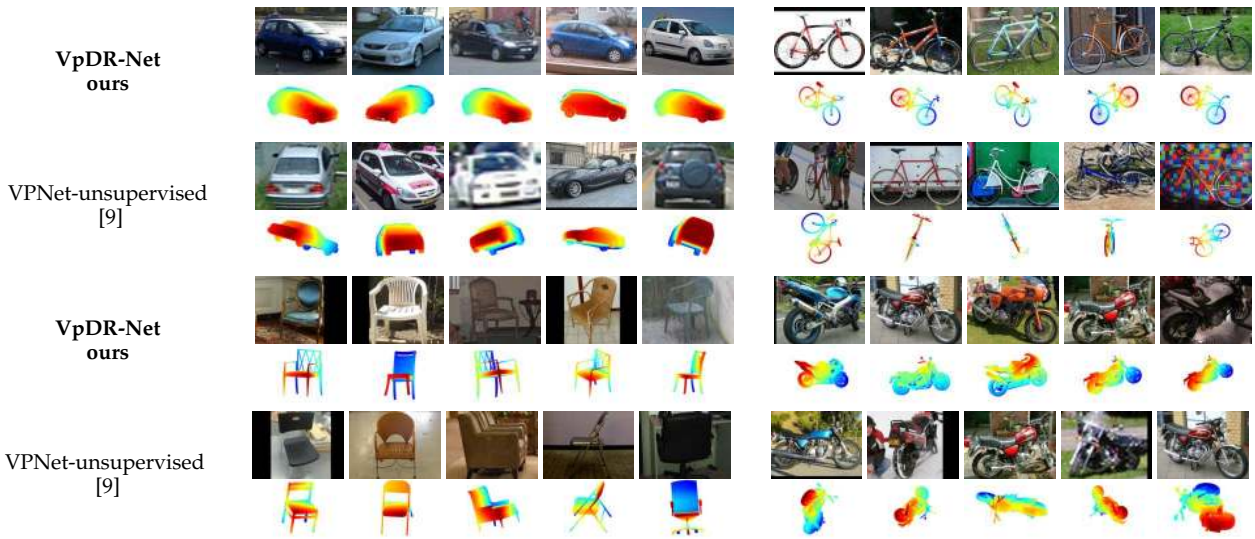

Fig. 8. **Viewpoint prediction.** Qualitative comparison between our VpDR-Net and the baseline VPNet architecture trained on Freiburg Cars / LDOS aligned with the method from [9] (VPNet-unsupervised). For each of the 4 considered object classes, the five most confident viewpoint predictions are visualized (sorted by the predicted confidence from left to right). Each predicted viewpoint is used to align the Pascal3D ground truth CAD model with the corresponding image.

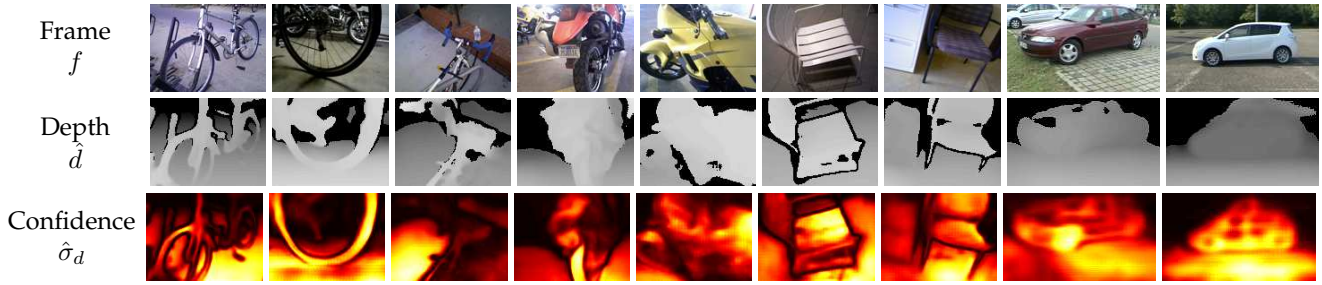

Fig. 9. **Monocular depth prediction.** Visualization of the predicted depth and confidence for different input images of the 4 considered classes. Depth maps are filtered by removing low confidence pixels. Lighter color corresponds to more confident regions.

	$\downarrow e_R$	$\downarrow e_C$	$\downarrow e_R^{rel}$	$\downarrow e_T^{rel}$	$\uparrow AP_{e_R}$	$\uparrow AP_{e_C}$
Test set: **LDOS** - chair						
VpDR-Net (ours)	**27.80**	**0.46**	**55.13**	**0.58**	**0.46**	**0.46**
VpDR-Net-NoAug	28.35	0.51	55.46	0.61	**0.46**	0.38
VpDR-Net-NoDepth	31.60	0.53	60.79	0.66	0.41	0.33
VpDR-Net-NoProb	68.74	0.83	85.78	0.89	0.05	0.06
Test set: **Pascal3D** - chair						
VpDR-Net (ours)	**33.70**	**14.23**	**57.61**	**0.72**	0.35	**0.34**
VpDR-Net-NoAug	33.96	15.26	67.12	0.79	**0.37**	0.29
VpDR-Net-NoDepth	35.34	18.78	68.68	0.85	0.30	0.14
VpDR-Net-NoProb	63.13	56.72	86.15	1.08	0.03	0.00

TABLE 2
Viewpoint prediction. Different flavors of VpDR-Net with removed components to evaluate their respective impact. All variations of VpDR-Net were trained on the LDOS videos of the chair class. ↑ (resp. ↓) means larger (resp. lower) is better.

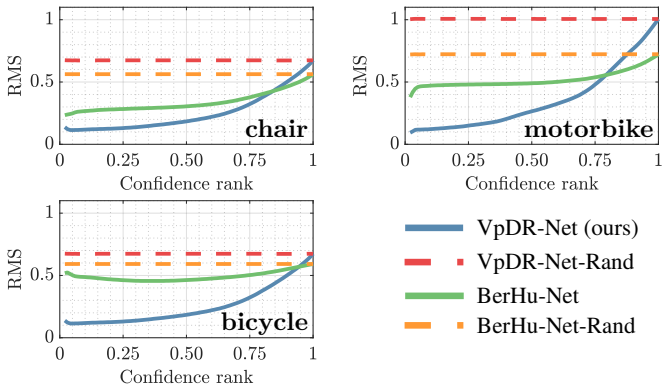

Fig. 10. **Monocular depth prediction.** Cumulative RMS depth reconstruction error for the LDOS data, when pixels are ranked by the predicted pixel-wise confidence.

aligned by a single global similarity $\mathcal{T}_G$ before assessment. The method for estimating $\mathcal{T}_G$ is detailed in sec. A.

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 $e_R^{\text{rel}} = 2^{-\frac{1}{2}} \| \ln R^*_{tt'} \hat{R}^\top_{tt'} \|_F$ and (2) the normalized relative translation error $e_T^{\text{rel}} = \|\hat{T}_{tt'} - T^*_{tt'}\|_2$, 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 [17], [19] 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 [64] with the final softmax classifier replaced by a viewpoint estimation layer that predicts the 6 DoF pose $\hat{g}_t^i$. Following [17], 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. In order to attach a confidence measure to these predictions, empirically we found that it was beneficial to use the average softmax value across the three max-scoring Euler angles.

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 [9]. In the fully-supervised setting, VPNet is trained by using ground-truth global camera poses provided by the Pascal3D training set.

Quantitative 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 [9] reports an error of 61.5 just for the azimuth component. Nevertheless, VpDR-Net outperforms VPNet-unsupervised in all the cases. The most significant difference in performance can be observed for the motorbike and bicycle classes. Here, the primary reason for the performance drop of VPNet-unsupervised is the inability of the alignment method from [9] to cope with an absence of the ground plane which is the case for the bicycle and motorbike point clouds. This shows the advantage of the proposed viewpoint factorization method compared to aligning 3D shapes as in [9]. Furthermore, the unsupervised VpDR-Net significantly reduces the gap with fully-supervised VPNet. We also 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-unsupervised, providing a reliable self-assessment mechanism. A qualitative comparison between VpDR-Net and the VPNet-unsupervised baseline are shown in Fig. 8.

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. 5.1.

We observe a significant performance drop when each of the components is removed. This confirms the importance of all contributions in the network design.

Additional experiments. We conducted more comparisons to the state of the art on the unaltered Pascal3D dataset, reporting the Average Viewpoint Precision (AVP) measure on the validation set as in [15]. Since AVP requires an object detector, we use the same set of RCNN [72] detections as in [17]. In order to estimate $\mathcal{T}_G$, we use the ground truth annotations from the training set. Due to the additional noise brought by the global alignment $\mathcal{T}_G$ we report results for the coarsest level of 4 orientation bins.

Method	AVP (4 bins) per object class			
	chair	bicycle	mbike	car
VpDR-Net (ours)	**16.6**	23.9	**30.1**	**33.4**
VPNet-unsupervised [9]	12.7	**24.5**	19.8	29.4
3D-DPM [73]	6.1	43.9	31.8	36.9
Vps & Kps [17]	25.1	59.4	61.1	55.2

TABLE 3
Joint viewpoint prediction and object detection on Pascal3D reporting the AVP measure on the validation set. 3D-DPM and Vps & Kps are fully supervised approaches while our VpDR-Net and the VPNet-unsupervised baseline do not require manual annotations.

The results are summarized in Table 3. VpDR-Net outperforms VPNet-unsupervised on 3 out of 4 classes while being comparable to the fully supervised 3D-DPM [73] on 3 out of 4 classes as well.

6.3 Depth prediction

We evaluate the monocular depth prediction module of VpDR-Net and in particular its ability to self-predict the quality of its prediction. These experiments are conducted on the test set of LDOS, since FrC does not contain ground-truth depth annotations.

The depth prediction 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 [30] based on the same Φ_{depth} subnetwork as VpDR-Net (but dropping Φ_{pcl} and Φ_{vp}). Following [30], 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 [74]. Finally, **BerHu-Net-Rand** is the same network, but predicting random confidence scores.

Quantitative results. Results are presented in Fig. 10, for the three LDOS categories. This figure shows the cumulative root-mean-squared (RMS) depth reconstruction error, after sorting pixels by their confidence as estimated by the network. We observe that, 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 on all considered classes. Our confidence mechanism is more effective in the case of motorbike and bicycle classes which is probably caused by the lower reliability of the ground truth signal obtained by using the IR depth sensor in a suboptimal outdoor setting.

Qualitative results. Fig. 9 shows qualitative results. In the case of chair, motorbike and bicycle depth predictions, we can observe higher uncertainty on the metallic surfaces (e.g. bicycle frames and legs of chairs) or areas lying on the boundaries of the objects. This is expected since the depth sensor provides erroneous signal in these cases. Similarly for

	↓ mD_{pcl}				↑ mVIoU			
Object class **dataset**	chair LDOS	bicycle LDOS	mbike LDOS	car FrC	chair LDOS	bicycle LDOS	mbike LDOS	car FrC
Aubry et al. [41]	0.49	0.69	0.84	0.41	0.04	0.04	0.04	0.21
VpDR-Net-Fuse (ours)	0.25	0.28	0.37	**0.23**	**0.14**	**0.12**	**0.13**	**0.29**
VpDR-Net (ours)	0.25	0.32	0.40	**0.23**	**0.14**	**0.12**	**0.13**	**0.29**
VpDR-Net-$\hat{P}_f$	0.43	0.53	0.71	0.56	0.09	0.09	0.05	0.11
VpDR-Net-$\hat{S}$	0.39	1.23	0.44	0.70	0.11	0.09	0.11	0.16
VpDR-Net-Chamfer	**0.19**	**0.23**	**0.32**	0.24	0.10	0.07	0.08	0.20

TABLE 4
Point cloud prediction. Comparison between different variants of VpDR-Net and the method of Aubry et al. [41].

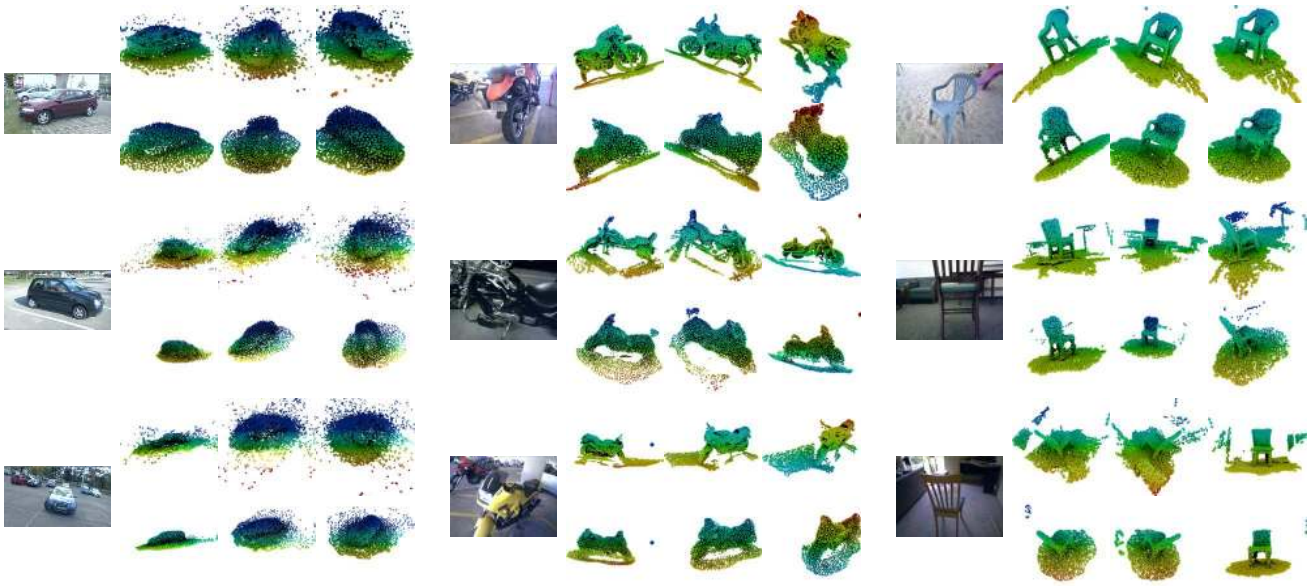

Fig. 11. **Point cloud prediction.** For every input image of an unseen object instance (left), ground-truth point cloud reconstruction using the full video sequence (top) and VpDR-Net point cloud prediction (bottom). For each reconstruction we show three different angles for better visualization.

the depth estimation of cars, the main modes of uncertainty can be observed on the specular areas (e.g. the bodywork) which is the case where ground truth providing multi-view stereo algorithm often fails.

6.4 Point cloud prediction

In this last set of experiments, we evaluate the point cloud completion module of VpDR-Net. The evaluation was conducted on the test sets of FrC and LDOS by comparing the predicted point clouds to the ground truth ones which were obtained as explained in sec. 6.1.

Evaluation measures. We use two evaluation measures: (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 [75]. We average these measures over the test set leading to mVIoU and mD_{pcl}. The normalized point cloud distance of [75] is computed as

$$D_{\text{pcl}}(C,\hat{C}) = \frac{1}{|C|}\sum_{c\in C}\min_{\hat{c}\in\hat{C}}\|\hat{c}-c\| + \frac{1}{|\hat{C}|}\sum_{\hat{c}\in\hat{C}}\min_{c\in C}\|\hat{c}-c\|.$$

For the VIoU measure, a voxel grid is setup around each ground truth point-cloud C by uniformly subdividing C's bounding volume into 30^3 voxels.

The point clouds are compared within the local coordinate frames of each frame's camera (whose focal length is assumed to be known). Furthermore, since the SfM reconstructions are known only up to a global scaling factor, we adjust each point cloud prediction $\hat{C}$ from the FrC dataset by multiplying it with a scaling factor ζ that aligns the means of $\hat{C}$ and C. Note that ζ can be computed analytically with:

$$\zeta = \frac{\mu_C^T\mu_{\hat{C}}}{\mu_{\hat{C}}^T\mu_{\hat{C}}}, \text{where}$$

$\mu_C = \frac{1}{|C|}\sum_{c_m\in C} c_m$ is the centroid of the point cloud C.

Baselines. VpDR-Net is compared against the approach of Aubry et al. [41] using their code. [41] 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 ground truth 3D point clouds. Then, given a testing image, [41] 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 our VpDR-Net, we evaluate several different flavors: the original VpDR-Net that predicts the point cloud $\hat{C}$, VpDR-Net-Fuse which further merges $\hat{C}$ with the predicted partial depth map point cloud $\hat{P}_f$, VpDR-Net-$\hat{P}_f$ which only predicts the partial point cloud $\hat{P}_f$, VpDR-Net-$\hat{S}$ that predicts the raw unfiltered and untruncated point cloud $\hat{S}$ and finally VpDR-Net-Chamfer which removes the density

predictions $\hat{\delta}$ and replaces $l_{pcl}(\hat{S})$ with a Chamfer distance loss as explained in [50].

Quantitative results. Table 4 shows that our reconstructions significantly outperform [41] on both metrics for both LDOS and FrC. Fusing the results with the original depth map produces a denser point cloud estimate and improves the results for some classes. The drops in performance by predicting solely the raw and partial point clouds $\hat{P}_f$ and $\hat{S}$ emphasize the importance of the point cloud completion and density prediction components respectively. The Chamfer distance loss brings marginal improvements in D_{pcl} but a significant decrease of VIoU due to the inability of the network to represent and discard outliers. Furthermore, as in [50], we have observed that the network tends to predict an average model of the object category with a limited amount of shape variation.

Qualitative results. Qualitative results are shown in Fig. 11. We can observe that in the case of chair and motorbike reconstructions, which are the classes with a large number of training videos and relatively clean ground truth point clouds, the reconstructions exhibit a large amount of details that allow to distinguish different geometric styles (e.g. an enduro vs a chopper). For the car reconstructions, where the number of training videos is lower and the ground truth point clouds are noisy due to erroneous SfM multi-view stereo depth, our model trades off statistical sensitivity for increased smoothness of the predictions.

7 Conclusion

In this work, we have considered the problem of predicting the 3D geometry of an object from a single image. We have demonstrated that motion cues can replace manual annotations and synthetic data for learning the geometry of object categories, and that the learned model successfully generalizes to new unseen instances, predicting the viewpoint, the depth and the shape of that new instance. Learning from motion cues is enabled by two innovations, a new image-based viewpoint factorization method and a new probabilistic shape representation, which we leveraged in a single neural network that simultaneously performs the three prediction tasks. As a third innovation, we have also demonstrated that allowing predictors to explicitly express uncertainty leads to significantly more robust learning. We validated our approach on four object categories demonstrating performance superior to existing approaches.

Acknowledgments. The authors gratefully acknowledge the support of NAVER LABS Europe and ERC 677195-IDIU.

References

[1] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proc. ECCV*, 1998.
[2] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.
[3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, pp. 105–112, 2011.
[4] 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*, vol. abs/1512.03012, 2015.
[5] 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.
[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.
[7] J. Carreira, S. Vicente, L. Agapito, and J. Batista, "Lifting object detection datasets into 3d," *PAMI*, vol. 38, no. 7, pp. 1342–1355, 2016.
[8] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3d object classes," in *Proc. CVPR*, 2009.
[9] N. Sedaghat and T. Brox, "Unsupervised generation of a viewpoint annotated car dataset from videos," in *Proc. ICCV*, 2015.
[10] D. Novotny, D. Larlus, and A. Vedaldi, "Learning 3d object categories by looking around them," in *Proc. ICCV*, 2017.
[11] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *Proc. ICCV*, 2007.
[12] M. Ozuysal, V. Lepetit, and P.Fua, "Pose estimation for category specific multiview object localization," in *Proc. CVPR*, 2009.
[13] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *Proc. ICCV*, 2011.
[14] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Multi-view priors for learning detectors from sparse viewpoint data," in *Proc. ICLR*, 2014.
[15] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *WACV*, 2014.
[16] R. Mottaghi, Y. Xiang, and S. Savarese, "A coarse-to-fine model for 3d pose estimation and sub-category recognition," in *Proc. CVPR*, 2015.
[17] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proc. CVPR*, 2015.
[18] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-Independent Object Class Detection using 3D Feature Maps," in *Proc. CVPR*, 2008.
[19] 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.
[20] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proc. CVPR*, 2017.
[21] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3d shape retrieval methods," *Multimedia Tools Appl.*, vol. 39, no. 3, pp. 441–471, Sep. 2008.
[22] T. Shen, H. Li, and X. Huang, "Approximately global optimization for robust alignment of generalized shapes," *PAMI*, vol. 33, pp. 1116–1131, 2010.
[23] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. CVPR*, 2014.
[24] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. NIPS*, 2014.
[25] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *PAMI*, vol. 38, no. 10, pp. 2024–2039, 2016.
[26] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *PAMI*, vol. 31, no. 5, pp. 824–840, 2009.
[27] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *PAMI*, vol. 36, no. 11, pp. 2144–2158, 2014.
[28] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs," in *Proc. CVPR*, 2015.
[29] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
[30] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016.
[31] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Proc. ECCV*, 2016.

[32] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. CVPR*, 2017.

[33] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. CVPR*, 2017.

[34] B. K. P. Horn, "Shape from shading." Cambridge, MA, USA: MIT Press, 1989, ch. Obtaining Shape from Shading Information, pp. 123–171.

[35] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *PAMI*, 2015.

[36] M. Prasad, A. Zisserman, and A. W. Fitzgibbon, "Single view reconstruction of curved surfaces," in *Proc. CVPR*, vol. 2, June 2006, pp. 1345–1354.

[37] E. Toppe, C. Nieuwenhuis, and D. Cremers, "Relative volume constraints for single view 3d reconstruction," in *Proc. CVPR*, 2013.

[38] L. G. Roberts, "Machine perception of three-dimensional solids," Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Electrical Engineering, 1963. [Online]. Available: http://www.packet.cc/files/mach-per-3D-solids.html

[39] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artif. Intell.*, vol. 31, no. 3, pp. 355–395, 1987.

[40] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing ikea objects: Fine pose estimation," in *Proc. ICCV*, 2013.

[41] 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.

[42] 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.

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

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

[45] 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.

[46] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proc. CVPR*, 2017.

[47] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation," *Proc. CVPR*, 2018.

[48] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3d object reconstruction," *3DV*, 2017.

[49] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani, "Surfnet: Generating 3d shape surfaces using deep residual networks," in *Proc. CVPR*, 2017.

[50] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proc. CVPR*, 2017.

[51] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *Proc. CVPR*, 2018.

[52] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *PAMI*, vol. 27, no. 3, pp. 418–433, 2005.

[53] M. Crocco, C. Rubino, and A. Del Bue, "Structure from motion with objects," in *Proc. CVPR*, June 2016.

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

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

[56] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3d structure from images," in *Proc. NIPS*, 2016.

[57] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proc. CVPR*, 2017.

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

[59] O. Wiles and A. Zisserman, "Silnet: Single-and multi-view reconstruction by learning from silhouettes," *Proc. BMVC*, 2017.

[60] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. CVPR*, 2016.

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

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

[63] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *RO*, vol. 33, no. 5, pp. 1255–1262, 2017.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

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

[66] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. CVPR*, 2017.

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

[68] T. J. Kozubowski, K. Podgórski, and I. Rychlik, "Multivariate generalized laplace distribution and related random fields," *Journal of Multivariate Analysis*, vol. 113, pp. 59–72, 2013.

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

[70] Y. Y. Morvan, "Acquisition, compression and rendering of depth and texture for multi-view video," Ph.D. dissertation, Technische Universiteit Eindhoven, 2009.

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

[72] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014.

[73] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," in *Proc. CVPR*, 2012.

[74] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," in *Proc. ICLR*, 2016.

[75] 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.

[76] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *PAMI*, vol. 13, no. 4, pp. 376–380, 1991.

David Novotny received the MS degree (with honors) in computer vision and machine learning from the Czech Technical University, Prague in 2015. He is currently a DPhil student in the VGG group, University of Oxford in collaboration with Naver Labs Europe. His current research interests are object detection, representation learning, matching, single-view 3D reconstruction and pose estimation.

Diane Larlus is a senior research scientist at NAVER LABS Europe. From 2005 to 2008, she worked as a doctoral candidate at INRIA Grenoble (getting a PhD degree in computer science in 2008) and did a robotics internship at the JRL/AIST laboratory in Tsukuba in summer 2007. From 2008 to 2010, she worked as a postdoc at TU Darmstadt and joined what has now become NAVER LABS Europe in 2010.

Andrea Vedaldi is assistant professor of Engineering Science at the University of Oxford, where he is co-PI in the Visual Geometry Group since 2012. He obtained his PhD degree at the computer science department of the University of California at Los Angeles in 2008, and for the BSc in Information Engineering at the University of Padova in 2003. He is currently sponsored by an ERC Starting Grant, and EPSRC Programme Grant, and a number of industrial grants and collaborations.

APPENDIX A

In this section, we detail the procedure from sec. 6.2 that estimates the global alignment transformation $\mathcal{T}_G$. Given a set of ground truth camera poses $g_i^* = (R_i^*, T_i^*)$ and the corresponding predictions $\hat{g}_i = (\hat{R}_i, \hat{T}_i)$, we want to estimate a global similarity transform $\mathcal{T}_G = (R_G, T_G, s_G)$, parametrized by a scale $s_G \in \mathbb{R}$, translation $T_G \in \mathbb{R}^3$ and rotation $R_G \in SO(3)$, such that the coordinate frames of g_i^* and $\hat{g}_i$ become aligned.

In more detail, the desired global similarity transform satisfies the following equation:

$$\hat{R}_i(R_G X + T_G) + s_G \hat{T}_i = R_i^* X + T_i^* \; ; \; \forall X \qquad (13)$$

i.e. given an arbitrary world-coordinate point $X \in \mathbb{R}^3$, its projection into the coordinate frame of g_i^* (the right part of eq. (13)) should be equal to the projection of X into the coordinate frame of $\hat{g}_i$ after transforming X with R_G, T_G and scaling the corresponding camera translation vector $\hat{T}_i$ with s_G (the left side of eq. (13)). Note that for LDOS data $\mathcal{T}_G$ corresponds to a rigid motion and $s_G = 1$. Given $\mathcal{T}_G$, the adjusted camera matrices $\hat{g}_i^{\text{adjust}}$ for which $\hat{g}_i^{\text{adjust}} \approx g_i^*$ are then computed with

$$\hat{g}_i^{\text{adjust}} = (\hat{R}_i R_G, \; \hat{R}_i T_G + s_G \hat{T}_i).$$

In order to estimate $\mathcal{T}_G$, X is substituted in eq. (13) with $X = C_i^* = -{R_i^*}^T T_i^*$, i.e. X is set to be the center of the ground truth camera g_i^* which is a valid point of the world coordinate frame. After performing some additional manipulations, we end up with the following constraint:

$$\forall i : \quad \frac{1}{s_G} R_G C_i^* + \frac{1}{s_G} T_G = \hat{C}_i, \qquad (14)$$

where $\hat{C}_i = -\hat{R}_i^T \hat{T}_i$ is the center of the predicted camera $\hat{g}_i$. Given the corresponding camera pairs $\{(g_i^*, \hat{g}_i)\}_{i=1}^N$ the constraint in eq. (14) is converted to a least squares minimization problem:

$$\underset{R_G, T_G, s_G}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \frac{1}{s_G} R_G C_i + \frac{1}{s_G} T_G - \hat{C}_i \right\|^2 \qquad (15)$$

and solved using the UMEYAMA algorithm [76].

For Pascal3D we estimate $\mathcal{T}_G$ from the held-out training set and later use it for evaluation on the test set. For LDOS, due to the absence of a held-out annotated training set, we estimate $\mathcal{T}_G$ on the test set.