Learning 3D Object Categories by Looking Around Them

Supplementary Material

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

A. Method: additional details

A.1. Scale ambiguity in SFM

In Sec. 3.2 in the paper, we explain that the scale ambiguity of structure from motion (SFM) causes each reconstruction of a sequence S^i to be known only up to a global sequence specific scaling factor λ^i. Since λ^i is not required to learn Φ_{vp}, but it is important for depth prediction (as discussed in Sec. 3.3 from the paper), 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 $\lambda^i_{t,t'} = \frac{\|T^i_{t'} - R^i_{t't} T^i_t\|}{\|\hat{T}^i_{t'} - R^i_{t't} \hat{T}^i_t\|}$. This expression allows us to conveniently estimate λ^i on the fly as a moving average during the SGD iterations used to learn Φ_{vp}, as samples $\lambda^i_{t,t'}$ can be computed essentially for free during this process.

A.2. The VpDR-Net architecture: further details

This section contains additional details about the layers that compose the different components of the VpDR-Net architecture.

The core architecture. The architecture of the VpDR-Net (introduced in Sec. 3.2 from the paper) is a variant of the ResNet-50 architecture [3] with some modifications to improve its performance as a viewpoint and depth predictor that we detail below.

In order to decrease the degree of geometrical invariance of the network, we first replace all 1×1 downsampling filters with full 2×2 convolutions. We then attach bilinear upsampling layers that first resize features from 3 different layers of the architecture (res2d, res3d, res4d) into fixed-size tensors and then sum them in order to create a multiscale intermediate image representation which resembles hypercolumns (HC) [2]. An extension of Fig. 2 from the paper that contains the diagram of this HC module can be found in Figure A.

Architecture of the viewpoint factorization network Φ_{vp}. 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. Figure C contains an overview of the viewpoint estimation module together with a detailed illustration of the modified downsampling residual blocks.

Architecture of the depth prediction Φ_{depth}. The depth prediction network (introduced in Sec. 3.3 from the paper) 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 [5]. More precisely, after attaching 2 standard residual blocks to the HC layers, the network also contains two 2x2 up-projection layers from [5] leading to a 64-dimensional representation of the same size as the input image. This is followed by 1x1 convolutional filters that predict the depth and confidence maps $\hat{D}_t$ and $\hat{\sigma}_{d_j}$ respectively. Figure B contains an illustration of Φ_{depth}.

Architecture of the point cloud completion network Φ_{pcl}. Differently from the two previous networks, the point cloud completion network Φ_{pcl} (introduced in Sec. 3.4 from the paper) is not convolutional but uses a residual multi-layer perceptron (MLP), *i.e.* a sequence of residual fully connected layers.

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:

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

The intermediate pooling operator, which is permutation in-

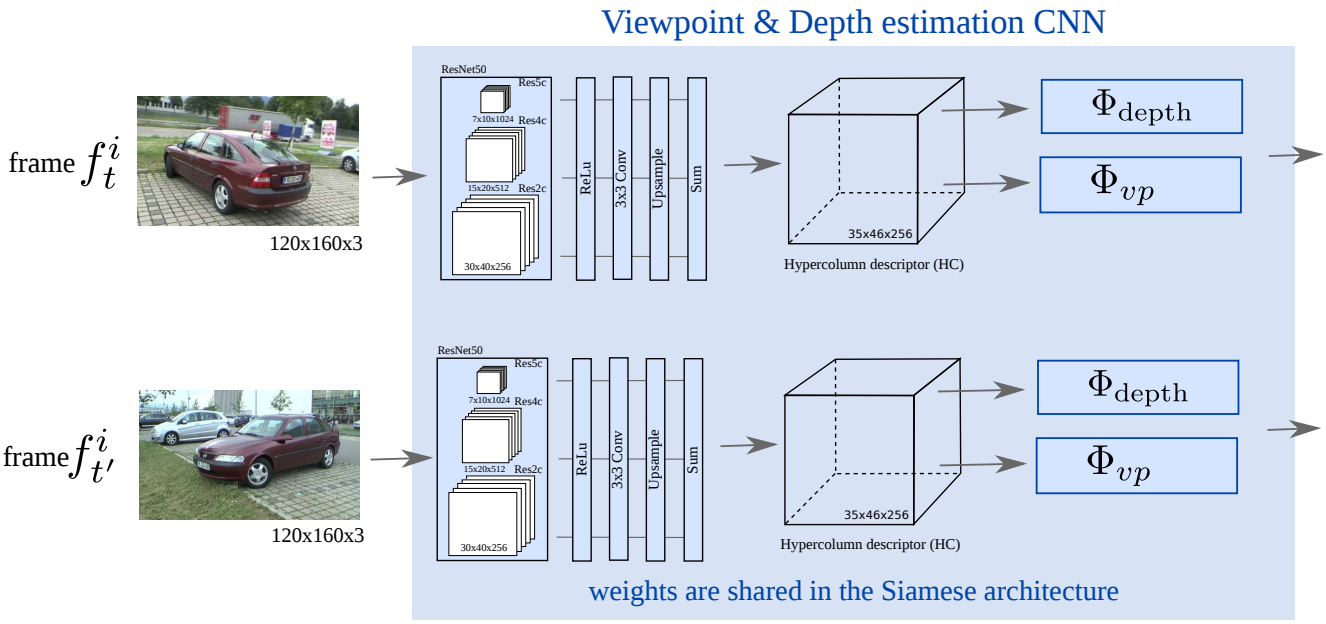

Figure A. **The core architecture of VpDR-Net.** This figure extends the Viewpoint & Depth estimation block from Figure 2 in the paper and describes the architecture of the hypercolumn (HC) module.

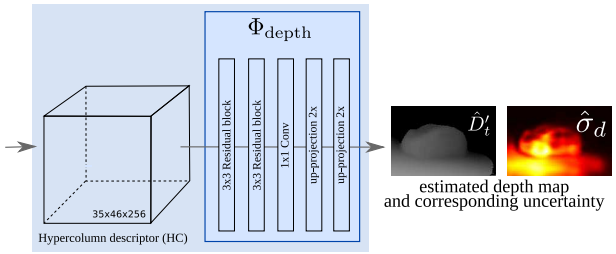

Figure B. **The architecture of Φ_{depth}.**

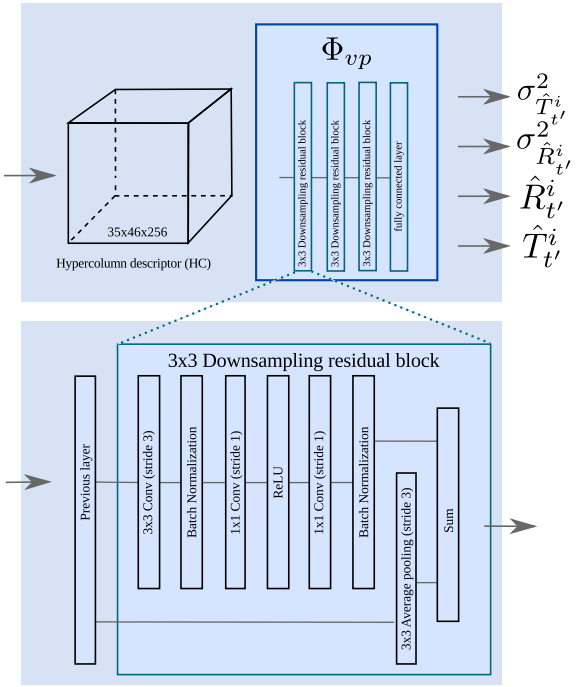

Figure C. **The architecture of Φ_{vp}.** Top: the layers of Φ_{vp}, bottom: A detail of the 3x3 downsampling residual block.

variant, removes the dependency on the number and order of input points $\hat{P}_f^G$. In practice, the pooling operator uses both max 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 differentiable bilinear sampling. Note that, following [10], 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 Figure D.

B. Experimental evaluation

In this section we provide additional details about the learning procedures of the baseline networks and about the experimental evaluation.

B.1. Learning details of BerHu-Net and VPNet

In this section we provide learning details for the BerHu-Net and VPNet baselines. The learning rates and batch sizes were in all cases adjusted empirically such that the conver-

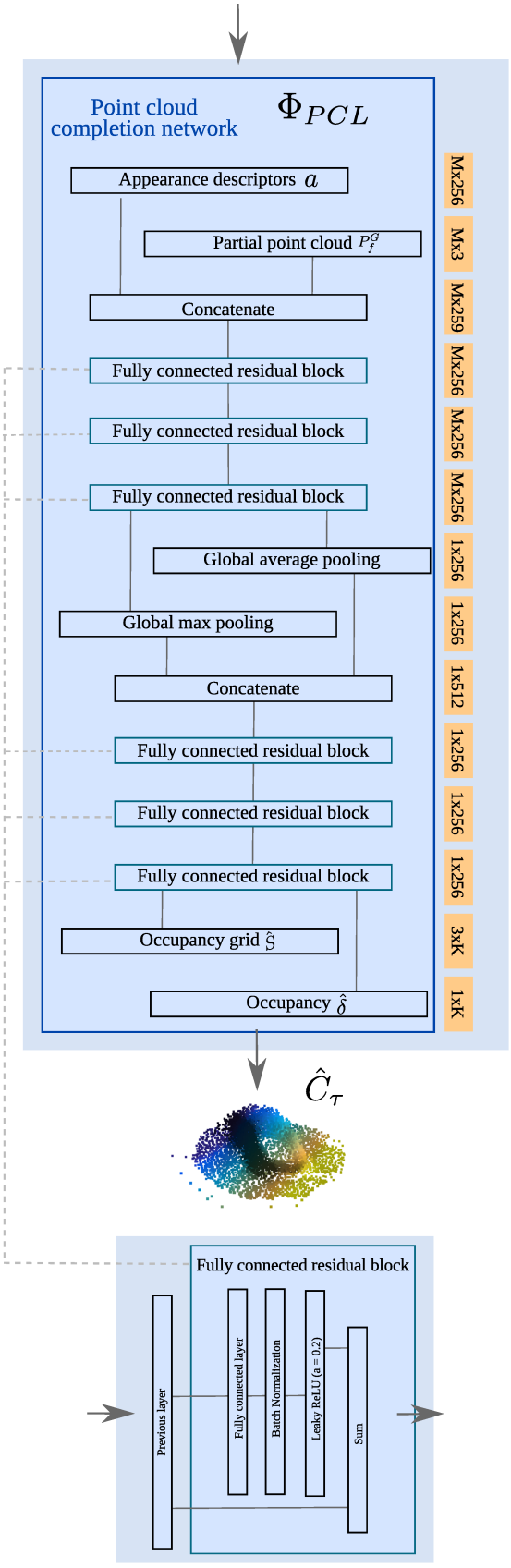

Figure D. **The architecture of Φ_{pcl}.** Top: The overview of the point cloud completion network, bottom: A detail of the fully connected residual block. Orange boxes denote the sizes of the layer outputs.

gence is achieved on the respective training sets.

BerHu-Net is trained with stochastic gradient descent with a momentum of 0.0005, initial learning rate 10^{-3} and a batch size of 16. The learning rate was lowered tenfold when no further improvement in the training losses was observed. The BerHu loss uses the adaptive adjustment of the loss cut-off threshold as explained in [5]. For the 2x2 up-projection layers we used the implementation of [5]. For each test image, we repeat the depth map extraction 70 times[1] with the dropout layer turned on and compute the variance of the predictions in order to obtain the per-pixel depth confidence values. The final feed-forward pass turns off the dropout layer and produces the actual depth predictions.

VPNet is trained with stochastic gradient descent with a momentum of 0.0005, initial learning rate 10^{-2} and a batch size of 128. The learning rate was lowered tenfold when no further improvement in the training losses was observed. For VPNet trained on aligned FrC, we adjusted the produced bounding box and viewpoint annotations in the same fashion as done for adjusting the Pascal3D annotations in sec. 5.1. in the paper, ensuring that the aligned FrC dataset is as compatible as possible with the target Pascal3D dataset. For LDOS, the produced dataset was adjusted in the same way except that we did not use the bounding boxes predicted by [9] because the input video frames already focus on full/truncated views of the object category.

B.2. Additional results

In sec. 5.1. in the paper we compared VpDR-Net to [9] on an adjusted version of the Pascal3D dataset. In this section, we additionally report the standard AVP measure [13] on the original Pascal3D dataset in order to present a better comparison with fully supervised state-of-the-art on this dataset. Because the AVP measure requires an object detector, we extract viewpoints from the same set of RCNN detections as in [11]. Due to the fact that the AVP measure, as well as most other measures from sec. 5.1. in the paper, depends on the dataset-specific global alignment transformation $\mathcal{T}_G$, we estimate it from the ground truth annotations of the training set of [13] using the same method as described in sec. 5.1. in the paper.

Due to the additional measurement noise brought by the estimation of $\mathcal{T}_G$, we report results only for the coarsest resolution of 4 azimuth bins. Our VpDR-Net obtained 33.4 and 14.7 AVP for the car and chair classes vs. 29.4 and 14.3 AVP obtained by [9] using the same detections from [11]. Our approach performs on par with some fully supervised approaches such as 3D DPM [7], while being inferior to the fully supervised state-of-the-art by the same margin as for the other metrics reported in table 1 in the paper.

[1] We empirically verified that 70 repetitions are enough for convergence of the variance estimates.

B.3. Absolute pose evaluation protocol

As noted in the paper, the absolute pose error metrics e_R and e_C can be computed only after aligning the implicit global coordinate frames of the benchmarked network and of the ground truth annotations. This procedure is explained in detail below.

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:

$$\hat{R}_i(R_G X + T_G) + s_G \hat{T}_i = R_i^* X + T_i^* \; ; \; \forall X \qquad (1)$$

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. (1)) 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. (1)). 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 $\hat{g}_i^{ADJUST}$ for which $\hat{g}_i^{ADJUST} \approx g_i^*$ are then computed with

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

In order to estimate $\mathcal{T}_G$, X is substituted in eq. (1) 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:

$$\frac{1}{s_G} R_G C_i^* + \frac{1}{s_G} T_G = \hat{C}_i \; ; \; \forall i \qquad (2)$$

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. (2) is converted to a least squares minimization problem:

$$\arg \min_{R_G, T_G, s_G} \sum_{i=1}^{N} || \frac{1}{s_G} R_G C_i + \frac{1}{s_G} T_G - \hat{C}_i ||^2 \qquad (3)$$

and solved using the UMEYAMA algorithm [12].

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.

B.4. Point cloud prediction

The normalized point cloud distance of [8] 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.

Test set	LDOS		FrC	
Metric	↑ mVIoU	↓ mD_{pcl}	↑ mVIoU	↓ mD_{pcl}
Aubry [1]	0.06	1.30	0.21	0.41
VpDR-Net-$\hat{P}_f$	0.10	0.37	0.11	0.56
VpDR-Net-Chamfer	0.09	**0.18**	0.20	**0.24**
VpDR-Net-$\hat{S}$	0.12	0.27	0.18	0.50
VpDR-Net (ours)	**0.13**	0.20	0.24	0.28
VpDR-Net-Fuse (ours)	**0.13**	0.19	**0.26**	0.26

Table A. **Point cloud prediction ablative study**. Comparison between VpDR-Net and the method of Aubry *et al.* [1] and an additional ablative study.

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:

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

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

Ablative study. In table 2 in the paper, we have presented a comparison of VpDR-Net to the baseline approach from [1]. Here we provide an additional ablative study that evaluates the contribution of the components of Φ_{pcl}. More exactly, table A extends table 2 from the paper with the following flavours of VpDR-Net: (1) VpDR-Net-$\hat{P}_f$ which only predicts the partial point cloud P_f, (2) VpDR-Net-Chamfer which removes the density predictions $\hat{\delta}$ and replaces $l_{pcl}(\hat{S})$ with a Chamfer distance loss and (3) VpDR-Net-$\hat{S}$ that predicts the raw unfiltered and untruncated point cloud $\hat{S}$.

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.

Related methods. Note that apart from [1], there exist newer works that tackle the problem of single-view 3D reconstruction [4, 6], However these were not considered due to their requirement of renderable mesh models which are not available in our supervision setting.

C. Qualitative results

Figures E to H contain additional viewpoint estimation results on Pascal3D. Differently from fig. 4 in the paper that was showing the most confident results, here we show *randomly selected* results of both VpDR-Net and the VP-Net which was trained on the corresponding aligned dataset (FrC or LDOS). Please refer to the captions for further details.

Additionally, in fig. I we provide qualitative comparisons of depth predictions between VpDR-Net and BerHu-Net on randomly selected images from the test set of LDOS. For an improved visualization, only the 80 % most confident pixel depth predictions are shown in each image, based on the confidence estimated by each model.

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. 4

[2] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. CVPR*, 2015. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1

[4] Q. Huang, H. Wang, and V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)*, 34(4):87, 2015. 4

[5] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 1, 3

[6] F. Massa, B. C. Russell, and M. Aubry. Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proc. CVPR*, 2016. 4

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

[8] 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. 4

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

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

[11] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. CVPR*, 2015. 3

[12] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *PAMI*, 13(4):376–380, 1991. 4

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

Figure E. **Viewpoint prediction on Pascal3D for VpDR-Net (ours)** on the car class. We show 40 randomly selected predictions from the test set sorted in descending order according to the predicted confidence scores. The images are sorted along the rows from left to right and from top to bottom, *i.e*. the most confident viewpoint is in the top left corner while the least confident image resides in the bottom right corner.

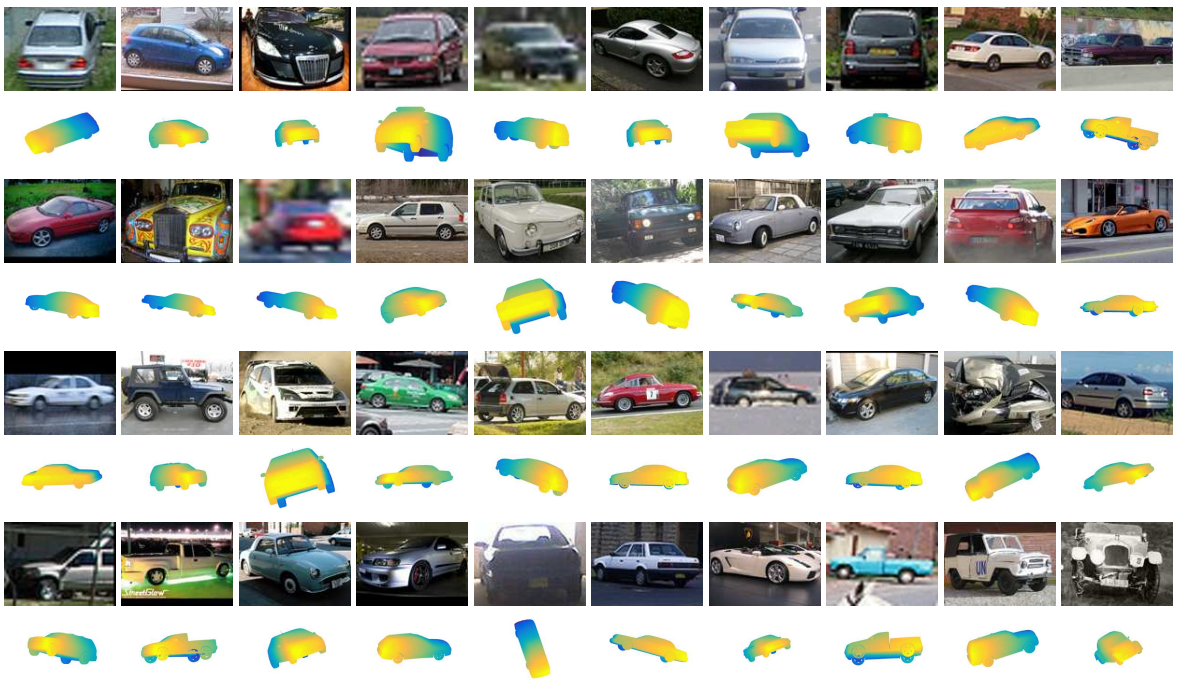

Figure F. **Viewpoint prediction on Pascal3D for VPNet (baseline) trained on aligned FrC** on the car class. We show 40 randomly selected predictions from the test set sorted in descending order according to the predicted confidence scores. The images are sorted along the rows from left to right and from top to bottom, *i.e*. the most confident viewpoint is in the top left corner while the least confident image resides in the bottom right corner.

Figure G. **Viewpoint prediction on Pascal3D for VpDR-Net (ours)** on the chair class. We show 40 randomly selected predictions from the test set sorted in descending order according to the predicted confidence scores. The images are sorted along the rows from left to right and from top to bottom, *i.e*. the most confident viewpoint is in the top left corner while the least confident image resides in the bottom right corner.

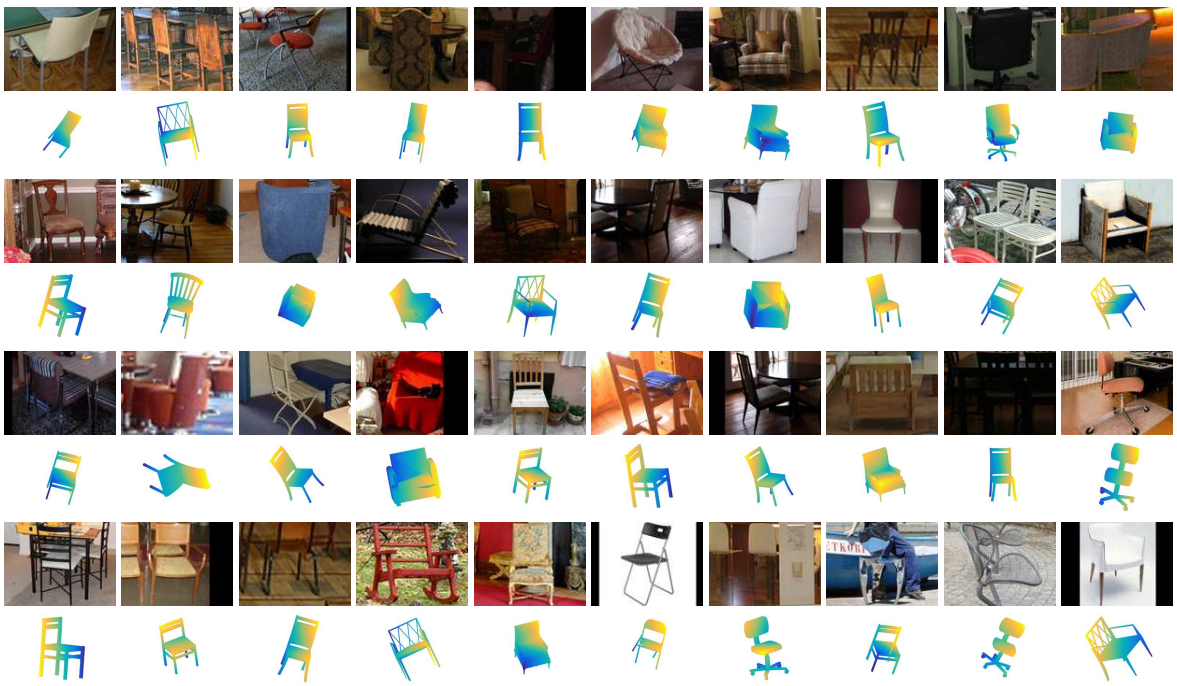

Figure H. **Viewpoint prediction on Pascal3D for VPNet (baseline) trained on aligned LDOS** on the chair class. We show 40 randomly selected predictions from the test set sorted in descending order according to the predicted confidence scores. The images are sorted along the rows from left to right and from top to bottom, *i.e*. the most confident viewpoint is in the top left corner while the least confident image resides in the bottom right corner.

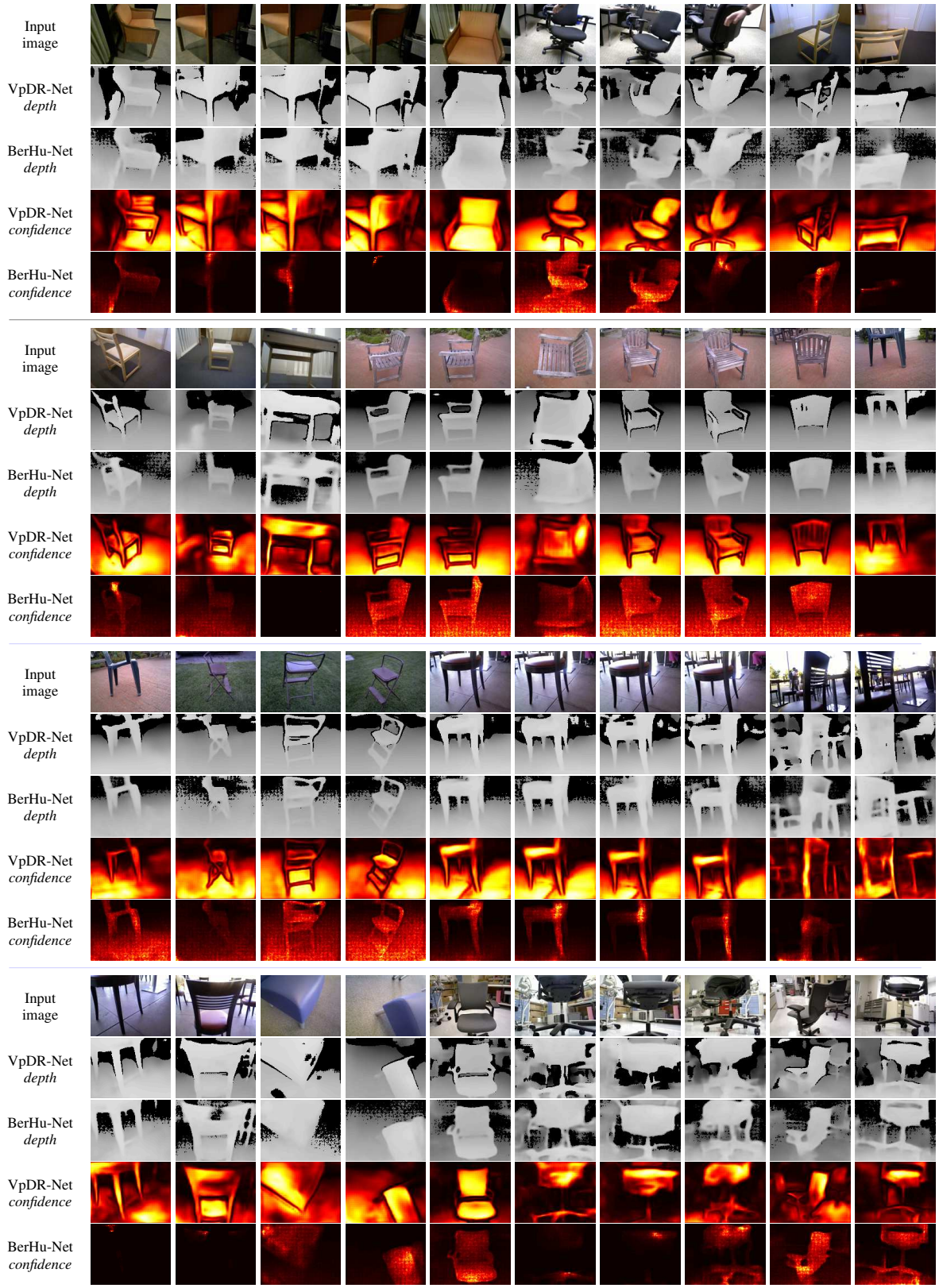

Figure I. **Depth prediction on random images from LDOS** comparing the predicted depth values as well as the predicted depth confidence of VpDR-Net (ours) and BerHu-Net.