Appendix : Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency

Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, Jitendra Malik
University of California, Berkeley
{shubhtuls, tinghuiz, efros, malik}@eecs.berkeley.edu

A1. Gradient Derivations

We re-iterate the equations for event probabilities and the ray consistency loss as defined in the main text.

$$p(z_r = i) = \begin{cases} (1 - x_i^r) \prod_{j=1}^{i-1} x_j^r, & \text{if } i \leq N_r \\ \prod_{j=1}^{N_r} x_j^r, & \text{if } i = N_r + 1 \end{cases} \tag{1}$$

$$L_r(x) = \sum_{i=1}^{N_r+1} \psi_r(i)\, p(z_r = i) \tag{2}$$

Expanding Eq. 2 using Eq. 1, we can get –

$$\begin{aligned} L_r(x) &= \sum_{i=1}^{N_r} \psi_r(i)\, (1 - x_i^r) \prod_{j=1}^{i-1} x_j^r \;+\; \psi_r(N_r + 1) \prod_{j=1}^{N_r} x_j^r \\ &= \sum_{i=1}^{N_r+1} \psi_r(i) \prod_{j=1}^{i-1} x_j^r \;-\; \sum_{i=1}^{N_r} \psi_r(i) \prod_{j=1}^{i} x_j^r \\ &= \psi_r(1) + \sum_{i=1}^{N_r} \psi_r(i+1) \prod_{j=1}^{i} x_j^r \;-\; \sum_{i=1}^{N_r} \psi_r(i) \prod_{j=1}^{i} x_j^r \end{aligned}$$

Simplifying this, we finally obtain –

$$L_r(x) = \psi_r(1) + \sum_{i=1}^{N_r} (\psi_r(i+1) - \psi_r(i)) \prod_{j=1}^{i} x_j^r \tag{3}$$

We can now compute the derivatives of the ray consistency loss w.r.t. the predictions x –

$$\begin{aligned} \frac{\partial\, L_r(x)}{\partial\, x_k^r} &= \sum_{i=1}^{N_r} (\psi_r(i+1) - \psi_r(i)) \frac{\partial \prod_{j=1}^{i} x_j^r}{\partial\, x_k^r} \\ &= \sum_{i=k}^{N_r} (\psi_r(i+1) - \psi_r(i)) \prod_{1 \leq j \leq i, j \neq k} x_j^r \end{aligned}$$

A2. Additional Discussion

A2.1. Formulation

Relation with Reprojection Error for Mask Supervision. In the scenario with foreground mask supervision, we had defined the corresponding cost term as in Eq. 4.

$$\psi_r^{mask}(i) = \begin{cases} s_r, & \text{if } i \leq N_r \\ 1 - s_r, & \text{if } i = N_r + 1 \end{cases} \tag{4}$$

Here, $s_r \in \{0, 1\}$ denotes the known information regarding each ray (or image pixel) - $s_r = 0$ implies the ray r intersects the object *i.e.* corresponds to an image pixel with foreground label, $s_r = 1$ indicates a pixel with background label. We observe that using this definition of event cost function, we can further simplify the loss defined in Eq. 3.

$$\begin{aligned} L_r^{mask}(x) &= \psi_r(1) + \sum_{i=1}^{N_r} (\psi_r(i+1) - \psi_r(i)) \prod_{j=1}^{i} x_j^r \\ &= s_r + \Big(\sum_{i=1}^{N_r-1} (s_r - s_r) \prod_{j=1}^{i} x_j^r \Big) + (1 - 2s_r) \prod_{i=1}^{N_r} x_i^r \\ &= s_r + (1 - 2s_r) \prod_{i=1}^{N_r} x_i^r \\ &= |\prod_{i=1}^{N_r} x_i^r - s_r|, \quad \text{if } s_r \in \{0, 1\} \end{aligned}$$

Therefore, we see that in the case of foreground mask observations, we minimize the discrepancy between the observation for the ray *i.e.* s_r and the reprojected occupancies ($\prod_{i=1}^{N_r} x_i^r$). This is similar to concurrent approaches designed to specifically use mask supervision for object reconstruction where a learned [3] or fixed [4] reprojection function is used. In particular, the reprojection function ($\min_{i=1}^{N_r} x_i^r$) used by Yan *et al.* [4] can be thought of as an approximation of the one derived using our formulation.

Event Costs for $z_r = N_r + 1$. We noted that to instantiate the event cost functions, we need to define the induced observations for the event of the ray escaping. While this is simple for some scenarios *e.g*. mask supervision, we need to define the induced observations ($d^r_{N_r+1}, p^r_{N_r+1}$ *etc*.) in other cases more carefully. When using depth observations for object reconstruction, we defined $d^r_{N_r+1}$ to be a fixed large value (10m as opposed to ∞) as we were penalizing absolute difference. Similarly, we used a uniform distribution as $p^r_{N_r+1}$ when using semantics observation so that the log-likelihood error is bounded. When using RBG supervision, we leveraged the knowledge that renderings had a white background to define $p^r_{N_r+1}$ – one could argue this is akin to implicitly using mask supervision but note that a) some models can also have white texture, b) our RGB-supervised model learned concavities which pure mask supervision could not, and c) we recover additional information (color) per voxel.

We wish to highlight here the takeaway that to completely define an event cost function, one must specify how the event of ray escaping is handled. While here we used a fixed induced observation corresponding to ray escaping events, this could equivalently be predicted in addition to the 3D shape *e.g*. predicting an environment map which allows us to lookup the color for an escaping ray conditioned on the direction can allow modelling the 3D shape and background with multi-view RGB observations.

Application Scope. We show four kinds of observations that can be leveraged via our framework – masks, depth, color and semantics. An obvious question is whether we can characterize scenarios the where the presented formulation may be applicable. As the main text notes, one fundamental assumption is the availability of relative pose between the prediction frame and observations. An aspect that is harder to formalize is what kinds of 'observations' can be used for supervision. We note that the only requirement is that an event cost function $\psi_r(i, [p^r_i])$ be defined such that it is differentiable w.r.t p^r_i. This implies that the known ray observation o_r should be 'explainable' given the ray (*i.e*. origin and direction), the location of the i^{th} voxel on its path, and some (view, ray agnostic) auxiliary prediction for the voxel.

A2.2. Experimental Setup

Biases in 'Ground-truth' Voxelization. We would like to mention a subtle practical aspect of benchmarking multi-view supervised approaches. The 'ground-truth' voxelization used for evaluation is binary. To obtain this binary label, typical voxelization methods (including the one we used) consider a grid cell occupied if there is any part of surface inside it. This is notably different from the information that view supervised approaches would obtain – a partially occupied cell would receive both sets of evidences as some rays would pass through it and some would not. Therefore, the binary 'ground-truth' used for evaluation is biased to be overinflated. It should be noted that the 3D supervised approaches have this bias in the supervision (as they use voxelizations computed similarly as ground-truth) whereas multi-view supervised approaches do not – we think this, at least partially, explains the gap between our multi-view supervised approach when using large number of views and 3D supervision. Note that previous mask-supervised approaches *e.g*. Yan *et al*. [4] did not suffer form this as they do not use actual rendered images for supervision but instead just use projections of the (biased) voxelizations as ground-truth mask 'views'. While in this work we use an IoU based metric, a metric that can handle a soft ground-truth might be more appropriate.

On Benchmarking Protocols for PASCAL 3D Dataset. Another subtle aspect we would like to highlight is that the 'ground-truth' 3D annotations on PASCAL3D dataset are obtained by choosing the appropriate model from a small set of CAD models. In particular, these models are shared for annotating train and test instances. Therefore, a benchmarking protocol where a learning based method has access to these models for training can bias the performance. To further highlight this point, we conducted a simple experiment. PASCAL3D has 65 CAD models across all categories (with 8, 10, 10 models respectively for aeroplanes, cars and chairs). Using the annotations on the training set, we train a classifier (by finetuning a pretrained ResNet-18 model) to choose the annotated model corresponding to an object given a cropped bounding box image. We then 'reconstruct' a test set object by retrieving the predicted model. We obtain IoUs of $(0.74, 0.76, 0.46)$ respectively for aeroplane, car and chair categories – significantly higher than our approach. Across the 10 object categories, the mean IoU via retrieval is 0.72.

The point we would like to emphasize is that using the small set of CAD models for both, training and testing will reward a learning system for biasing itself towards these small set of models and is therefore not a recommended benchmarking protocol. Note that since we train using masks and pose for supervision, we do not leverage these models for training and only use them for evaluation.

A3. CNN Architectures, Experiment Details and Result Interpretation

For all the object-reconstruction experiments, similar to previous learning based methods [1, 2], our CNN is trained to output reconstructions in a canonical frame (where object is front-facing and upright) – the known camera associated with each observation image is assumed w.r.t. this canonical frame. For the scene reconstruction experiment, we output the reconstructions w.r.t. the frame associated with the

camera corresponding to the input image. We therefore assume known relative transformations between the input image frame and the used observations.

As mentioned in the main text, to efficiently implement the view consistency loss between the prediction and an observation image, we randomly sample rays. We always sample total 3000 rays (chosen to keep iteration time under 1 sec) for each instance in the mini-batch *i.e*. if we have k observation images, we sample $\frac{3000}{k}$ rays per observation. Additionally, for efficiency in the scene reconstruction experiment, instead of using all 30 observations in the sequence, we randomly sample 3 observations per iteration (and sample 1000 rays per observation). For ShapeNet experiments, we use all available observations (typically 5) – for the ablation where we vary number the number of observations available, we use all k observations in each mini-batch. Finally, we observed that a large fraction of rays in the object reconstruction experiments (ShapeNet and PASCAL VOC) corresponded to background pixels. To counter this, we weight the loss for the foreground rays higher by a factor of 5 (hyperparameter chosen using chair reconstruction evaluation on the validation set using Mask observations on ShapeNet) – an alternate would be to simply sample more foreground rays.

In all the experiments, the basic network architecture is of an encoder-decoder form. The encoder takes as input an RGB image and uses 2D convolutions and fully-connected layers to encode it. The decoder performs 3D up-convolutions to finally predict the voxel occupancies (and optional per-voxel predictions). To succinctly describe the architectures, we denote by $C(n)$ a convolution layer with 3 X 3 kernel and n output channels, followed by a $ReLU$ non-linearity and spatial pooling such that the spatial dimensions reduce by a factor of two. We denote by $R()$ a reshape layer that reshapes the input according to the size specified when instantiating it. Similarly by $UC(n)$, we denote a 3D up-convolution followed by $ReLU$, where the output's spatial size is doubled. Unless otherwise specified, we initialize all weights randomly and use ADAM for training the networks.

A3.1. ShapeNet (Mask/Depth Supervision)

Architecture. The CNN architecture used here is extremely simple and has an encoder $Img(3, 64, 64) - C(8) - C(16) - C(32) - C(64) - C(128) - FC(100) - FC(100) - FC(128) - R(16, 2, 2, 2)$. The decoder is of the form $UC(8) - UC(4) - UC(2) - UC(1) - \text{Sigmoid}()$. The final voxel grid has 32 cells in each dimension. Note that since the aim here is to analyze the approach in a simple setting, we train a separate CNN per category, unlike the PASCAL VOC experiments where we follow previous work to train a category-agnostic CNN.

Rendered Data. To obtain the RGB/Depth/Mask images which are used as input for the CNN or as 'observations' for training using the view consistency loss, we render the ShapeNet models using Blender. For each CAD model, we render images from random viewpoints obtained via uniformly sampling azimuth and elevation from $[0, 360)$ and $[-20, 30]$ degrees respectively. We also use random lighting variations to render the RGB images.

Analysis. The experiments and ablations revealed some interesting trends. Our DRC approach is robust to observation noise and using about 4-5 observations saturates performance. While this performance is still below using ground-truth supervision, we conjecture this may be to due to the inflation bias in the 'ground-truth'. Also, using mask supervision and depth supervision are both equally informative when considering a large (more than 5) number of views though depth images are more informative when using 1-2 observations.

A3.2. PASCAL VOC Reconstruction

The decoder architecture used in these experiments is similar to the ShapeNet architecture described above. However, in the encoder, we use the convolution layers from ResNet-18 (initialized with copied weights from a pre-trained Imagenet classification network).

A3.3. Cityscapes Reconstruction

Architecture. Since the images in this dataset typically have a higher field of view in the x direction than the y direction, we use the architecture exactly same as the PASCAL VOC experiment (*i.e*. with ResNet layers) with the modification that the last two layers of the encoder are $FC(256) - R(16, 4, 2, 2)$. This allows the final voxel prediction to have more cells (64) in the x dimension. In addition to voxel occupancies, the last layer also outputs a per-voxel semantic class distribution (and applies a Softmax instead of Sigmoid to these).

Grid Parametrization. Since outdoor scenes have a much larger spatial extent, we use a non-uniform voxel grid in this experiment. The cell sizes increase as the z coordinate increases. This has the effect of modeling the nearby structures with more resolution but the far-away structures are captured only coarsely. The voxel-coordinate system lies in the range $x \in [0, 64], y \in [0, 32], z \in [0, 32]$ with the cell centres located at integer locations offset by $\frac{1}{2}$. The correspondence of a coordinate (x, y, z) of this system to the real world is given by $p = \alpha_1 e^{\alpha_2 z}(f(x - 32), f(y - 16), 1)$. Note that the volumetric bins are uniform when projected into an image. Here α_1, α_2 are defined s.t. the minimum and maximum z coordinates are 0.5m are 1000m. The parameter f is defined to give a horizontal field of view of 50 degrees.

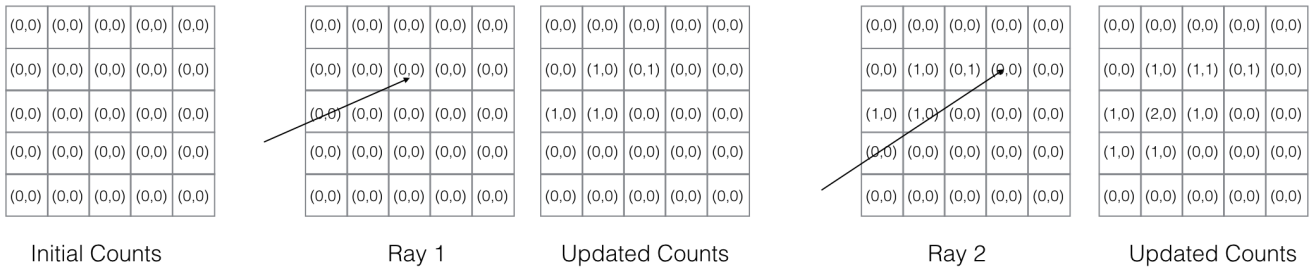

Figure 1: We show the process of precomputing a pseudo-groundtruth 3D model using available depth images (denoted as the 'Depth Fusion' baseline in main text). We maintain (empty,occupied) count for each voxel which respectively indicate the number of rays that pass through the voxel or terminate in it. Given the observed depth images, we can compute the starting and termination points of all rays. We show an example of passing two rays and updating the maintained counts. The final counts are used to determine a soft occupancy value for each voxel which serves as the prediction target – we ignore the voxels where both counts are 0.

A3.4. RGB-Supervised Reconstruction

Architecture. The CNN architecture here is the same as the one used for 3D prediction on ShapeNet with depth/mask supervision with the change that the last UC layer predicts 4 channels instead of 1 where the last three correspond to per-voxel color predictions.

Analysis. We observed that our learned model using multi-view color image observations as supervision could recover, from a single input image, the shape and texture of the full 3D shape including details like concavities in chairs, texture on unseen surfaces *etc*. An interesting error mode however, was a white cloud-like structure predicted below cars. This was predicted because, due to limited elevation variation in our view sampling, all the rays that pass through that region actually correspond to the background and so have a white color associated with them. This observation can be 'explained' via two solutions - a) to predict empty space and allow the rays to escape, and b) to predict occupied space with white texture. Our learned CNN chooses the latter way of explaining the multi-view observations.

A4. Depth Fusion

We show in Figure 1 the process of computing the pseudo-ground truth shape used for our baseline. Note that this is applicable only for depth images but not object mask observations since the termination points of the rays are unknown in case of foreground masks. This process is analogous to extracting and averaging unary terms for voxel occupancies for each ray – the voxels that rays pass through are assumed to be empty and the ones where they terminate occupied.

This process does not model any noise in observations unlike ours, which has a higher-order cost term. To clarify this, consider a ray for which we have a (noisy) depth observation. Let $v1, v2, v3$ be 3 voxels (in order) in its path and let the observed depth imply termination in $v3$. Suppose that the 'true' model has $v2$ occupied and not $v3$. Under our cost term, the 'true' shape would incur only a small cost for the noisy observation but under if we only have unary terms as in the fusion approach, this would incur a high cost.

References

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

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

[3] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. In *NIPS*, 2016. 1

[4] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016. 1, 2