Supplementary Material: MegaDepth: Learning Single-View Depth Prediction from Internet Photos

Zhengqi Li Noah Snavely
Department of Computer Science & Cornell Tech, Cornell University

In this supplementary material, we first describe additional details about our depth map refinement/enhancing algorithms (Section 1). We then include several additional qualitative comparisons to show the effects of $\mathcal{L}_{\text{grad}}$ and $\mathcal{L}_{\text{ord}}$ in Section 2. In Section 3, we provide additional details for of our SfM Disagreement Rate (SDR). Finally, in Section 4, we provide additional qualitative results in the form of depth maps predicted from images in our MegaDepth test set, as well as the Make3D, KITTI and Depth in the Wild (DIW) test sets.

1. Depth Map Refinement and Enhancement

In this section, we provide additional details for our depth map refinement and enhancement methods presented in Section 3.2 and 3.3 of the main paper.

1.1. Modified MVS algorithm

Our modified MVS algorithm and semantic segmentation-based depth map filtering are summarized in Algorithm 1. Our algorithm first runs PatchMatch [1] using photometric consistency constraints, as implemented in COLMAP, to solve for an initial depth map D^0 (with some pixels whose depth could not be estimated marked as invalid). Next, K iterations of PatchMatch using geometric consistency constraints are run. For each iteration k, we compare the depth values at each pixel before and after the update and keep the smaller (closer) of the two, to get an updated depth map D^k. After K iterations of PatchMatch, we apply a median filter to D^K and only keep depths whose values are stable, in that they are close to their median-filtered value. Finally, we remove spurious depths from transient objects based on semantic segmentation, as described in Section 3.3 of the main paper. Regarding the parameters defined in Algorithm 1, we set $\tau_1 = \tau_2 = 1.15$ and $K = 3$. Two additional examples of depth maps with and without our refinements are shown in Figure 1.

Algorithm 1 Depth Refinement and Semantic Cleaning

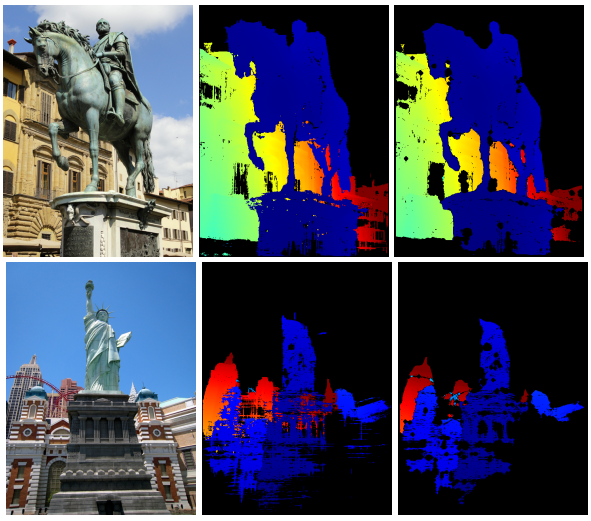
Input: Input image I, semantic segmentation map L (divided into subregions F (foreground), B (background), and S (sky).
Output: Refined depth map D for image I.

Run PatchMatch using photometric consistency constraints to solve for initial depth estimate of D^0. Pixels in D^0 without an assigned depth are instead assigned a `NaN` sentinel value.
for round $k = 1$ to K **do**
  Run PatchMatch using geometric consistency constraints on D^{k-1} to get updated depth estimate D^k.
  $R^k = D^k / D^{k-1}$ (element-wise)
  for each valid (non-`NaN`) pixel p of R^k **do**
    if $R_p^k > \tau_1$ **then**
      $D_p^k = D_p^{k-1}$
    else
      $D_p^k = D_p^k$
Apply 5×5 median filter on D^K, storing result in $\hat{D}^K$.
Filter (*i.e.*, replace with `NaN`) unstable pixels from D_p^K for which $\max(\hat{D}_p^K / D_p^K, D_p^K / \hat{D}^K) > \tau_2$.
for each connected component C from F **do**
  if fraction of valid depths in C is $> 50\%$ **then**
    keep depths in region C from D^K.
  else
    remove all depths in region C from D^K.
Filter out all depths in sky region S.
Apply morphological erosion followed by small connected components removal operation on D^K to obtain final depth map D.

1.2. Foreground and background classes

In this subsection, we provide details of the foreground object classes used to define the foreground mask F for each image, and similarly the background object classes used to define the background mask B. These classes are subsets of the classes recognized by our semantic segmentation module, as described in Section 3.3 of the main paper.

Foreground classes. F = {person, table, chair, seat, signboard, flower, book, bench, boat, bus, truck, streetlight, booth, poster, van, ship, fountain, bag, minibike, ball, animal, bicycle, sculpture, traffic light, bulletin board}



(a) Input photo (b) Raw depth (c) Refined depth

Figure 1: **Additional example comparisons between MVS depth maps with and without our proposed refinement/cleaning methods. Column (b) (before filtering)**: the plinth of the statue in the first row and the "Statue of Liberty" in the second row both show depth bleeding effect. **Column (c) (after filtering)**: our refinement method corrects or removes such depth values.

Background classes. B = {building, house, skyscraper, hill, tower, waterfall, mountain}.

1.3. Automatic ordinal depth labeling

In this subsection, we provide additional details for our automatic ordinal depth labeling method. Recall that O ("Ordinal") is the subset of photos that do *not* satisfy the "no selfies" criterion described in the main paper. Recall that the "no selfies" criterion rejects images I for which $< 30\%$ of the pixels (ignoring the sky region S) consists of valid depth values)—otherwise, these images are added to the set O. For each image $I \in O$, and given foreground pixel F and B in I as defined above, we compute two regions, $F_{\text{ord}} \in F$ and $B_{\text{ord}} \in B$, such that all pixels in F_{ord} are likely in front of all pixels in B_{ord}.

In particular, we assign any connected component C of F to F_{ord} if the area of C is larger than 5% of the image. We assign a pixel $p \in B$ to B_{ord} if it satisfies the following conditions:

1. p belongs to the background region B,
2. the area of p's connected component in B is larger than 5% of the image, and
3. p has a valid depth value that lies in the last quartile of the full range of depths for I.

Originally, we considered a more complex approach involving geometric reasoning (e.g., estimating where foreground objects touch the ground), but we found that the simple approach above works very well ($> 95\%$ accuracy in pairwise ordinal relationships), likely because natural photos tend to be composed in certain common ways. Additional examples of our automatic ordinal depth labels are shown in Figure 2.

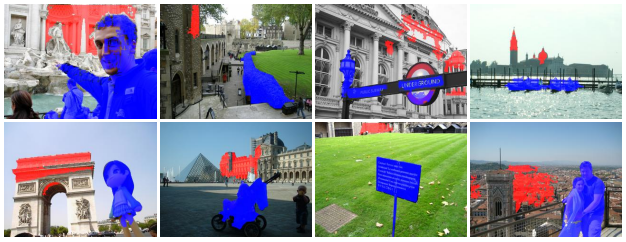


Figure 2: **Additional examples of automatic ordinal labeling.** Blue mask: foreground (F_{ord}) derived from semantic segmentation. Red mask: background (B_{ord}) derived from reconstructed depth.

2. Additional examples of the effects of $\mathcal{L}_{\text{grad}}$ and $\mathcal{L}_{\text{ord}}$

Here we show additional qualitative examples of the effects of our loss terms $\mathcal{L}_{\text{grad}}$ and $\mathcal{L}_{\text{ord}}$ effects on learned single-view depth predictions. Figure 3 shows the effect of $\mathcal{L}_{\text{grad}}$ on predicted depth maps, and Figure 4 shows the effect of $\mathcal{L}_{\text{ord}}$.

3. SfM Disagreement Rate (SDR)

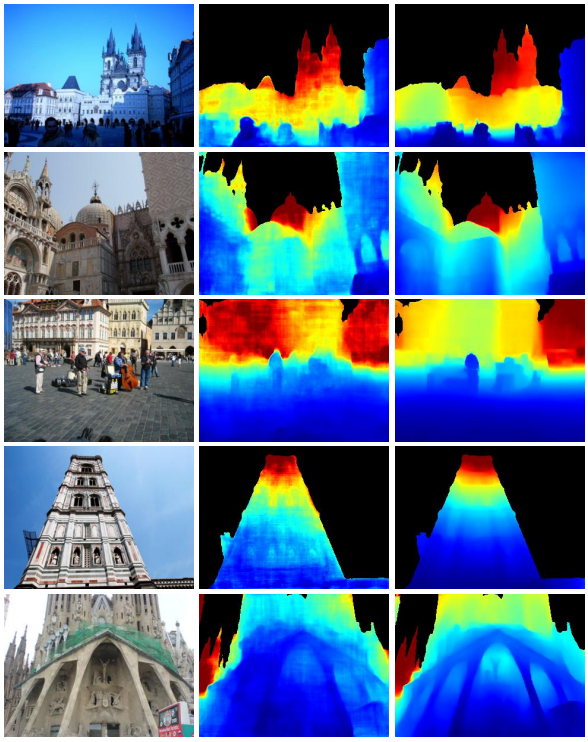
In this section, we provide additional details for our SfM Disagreement Rate (SDR) error metric defined in Section 5.1 of the main paper.

SDR is based on the rate of disagreement between a predicted depth map and the ordinal depth relationships derived from estimated ground truth SfM points. We use sparse SfM points for this purpose rather than dense MVS depths for two reasons: (1) we found that sparse SfM points can capture some structures not reconstructed by MVS (e.g., complex objects such as lampposts), and (2) we can select a robust subset of SfM points based on measures from SfM such as the number of observing cameras or uncertainty of the estimated depth computed by bundle adjustment.

We define $\text{SDR}(D, D^*)$, the ordinal disagreement rate between the predicted (non-log) depth map $D = \exp(L)$ and ground-truth SfM depths D^*, as:

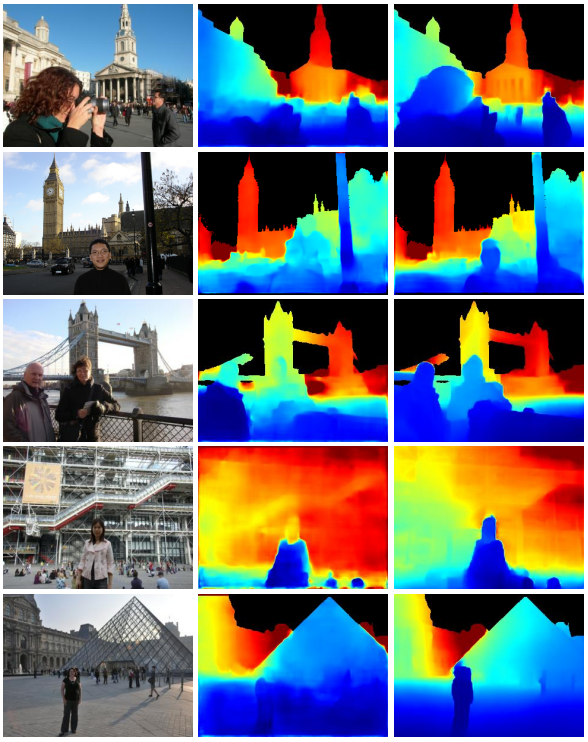
$$\text{SDR}(D, D^*) = \frac{1}{n} \sum_{i,j \in \mathcal{P}} \mathbb{1}\left(\text{ord}(D_i, D_j) \neq \text{ord}(D_i^*, D_j^*)\right) \quad (1)$$

where $\mathcal{P}$ is the set of pairs of pixels with available SfM depths to compare, n is the total number of pairwise comparisons, and $\text{ord}(\cdot, \cdot)$ is one of three depth relations (*further-*



Input photo | Output w/o $\mathcal{L}_{\text{grad}}$ | Output w/ $\mathcal{L}_{\text{grad}}$

Figure 3: **Depth predictions with and without $\mathcal{L}_{\text{grad}}$.** $\mathcal{L}_{\text{grad}}$ encourages the prediction to match the depth gradient of the ground truth.



Input photo | Output w/o $\mathcal{L}_{\text{ord}}$ | Output w/ $\mathcal{L}_{\text{ord}}$

Figure 4: **Depth predictions with and without $\mathcal{L}_{\text{ord}}$.** $\mathcal{L}_{\text{ord}}$ corrects ordinal depth relations for hard-to-construct objects such as people.

than, *closer-than*, and *same-depth-as*):

$$\text{ord}(D_i, D_j) = \begin{cases} 1 & \text{if } \frac{D_i}{D_j} > 1 + \delta \\ -1 & \text{if } \frac{D_i}{D_j} < 1 - \delta \\ 0 & \text{if } 1 - \delta \leq \frac{D_i}{D_j} \leq 1 + \delta \end{cases} \quad (2)$$

In other words, SDR is the rate of disagreement between predicted and ground-truth depths in terms of pairwise depth orderings. Note that SDR is an unweighted measure for simplicity (all measurements count the same towards the cost), but we can also integrate depth uncertainty derived from bundle adjustment as a weight.

We also define $\text{SDR}^{=}$ and $\text{SDR}^{\neq}$ as the disagreement rate with $\text{ord}(D_i^*, D_j^*) = 0$ and $\text{ord}(D_i^*, D_j^*) \neq 0$ respectively. In our experiments, we set $\delta = 0.1$ for tolerance to uncertainty in SfM points.

Because SDR is based on point pairs and hence takes $O(n^2)$ time to compute, for efficiency we subsample SfM points by splitting each image into 15×15 blocks, and for each block, randomly sampling an SfM point (if any exist). We then use these sampled points to create a clique of ordinal relations, where each edge connecting two features is augmented with the ordinal depth label. To obtain reliable sparse points we only sample SfM points seen by > 5 cameras and with reprojection error < 3 pixels. Figure 5 shows several examples of SfM points we sample for evaluating SDR.

4. Qualitative Results

In this section, we provide additional qualitative results on our MegaDepth test set, as well as the Make3D, KITTI and Depth in the Wild (DIW) test sets.

4.1. Qualitative results on MegaDepth test set

Qualitative results on images from the MegaDepth test set are shown in Figure 6. As in the main paper, we compare single-view depth prediction results from three network architectures: (1) VGG, using the same network and loss as [3], (2) ResNets, adopted from [5], and (3) the "hourglass"(HG) network adopted from [2].

4.2. Qualitative results on Make3D

Figure 7 shows qualitative comparisons between our MD-trained network and other non-Make3D dataset-trained networks on the Make3D test set. In particular, we compare our method with (1) a DIW-trained network [2], (2) the best NYU-trained network [3], and (3) the best KITTI-trained network [4]. None of the methods used Make3D data during training.

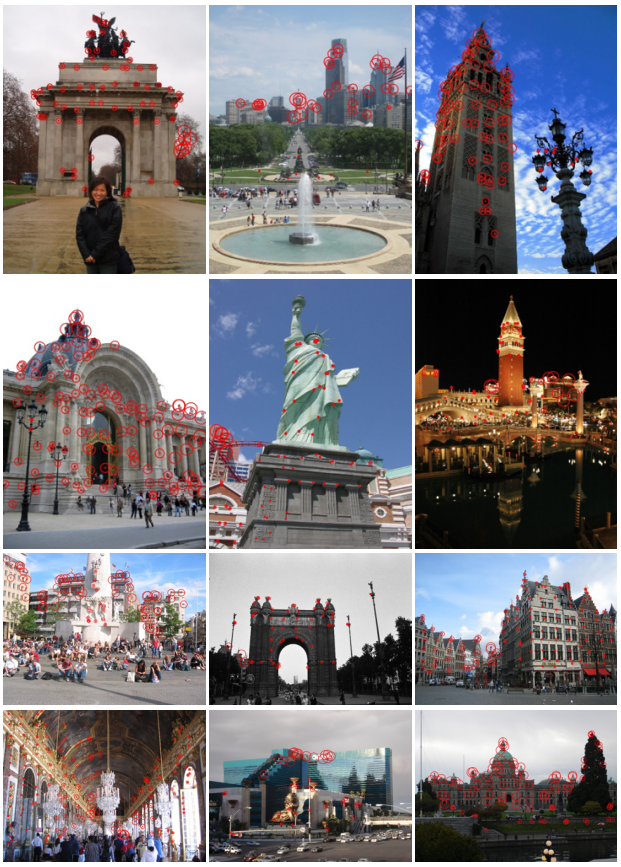


Figure 5: **Examples of sampled SfM points.** Red circles indicate sampled SfM points with the radius indicating estimated depth derived from SfM; small radius = small (close) depth, large radius = large (far) depth.

4.3. Qualitative results on KITTI

Figure 8 shows qualitative comparisons between our MD-trained network and other non-KITTI dataset-trained networks on the KITTI test set. In particular, we compare our method with (1) a DIW-trained network [2], (2) the best NYU-trained network [7], and (3) the best Make3D-trained network [5]. None of the methods used KITTI data during training.

4.4. Qualitative results on DIW

Figure 9 shows qualitative comparisons between our MD-trained network and other non-DIW dataset-trained networks on the DIW test set. In particular, we compare our method with (1) the best NYU-trained network [3] (2) the best KITTI-trained network [4], and (3) the best Make3D-trained network [6]. None of the methods used DIW data during training.

References

[1] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Proc. British Machine Vision Conf. (BMVC)*, 2011.

[2] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Neural Information Processing Systems*, pages 730–738, 2016.

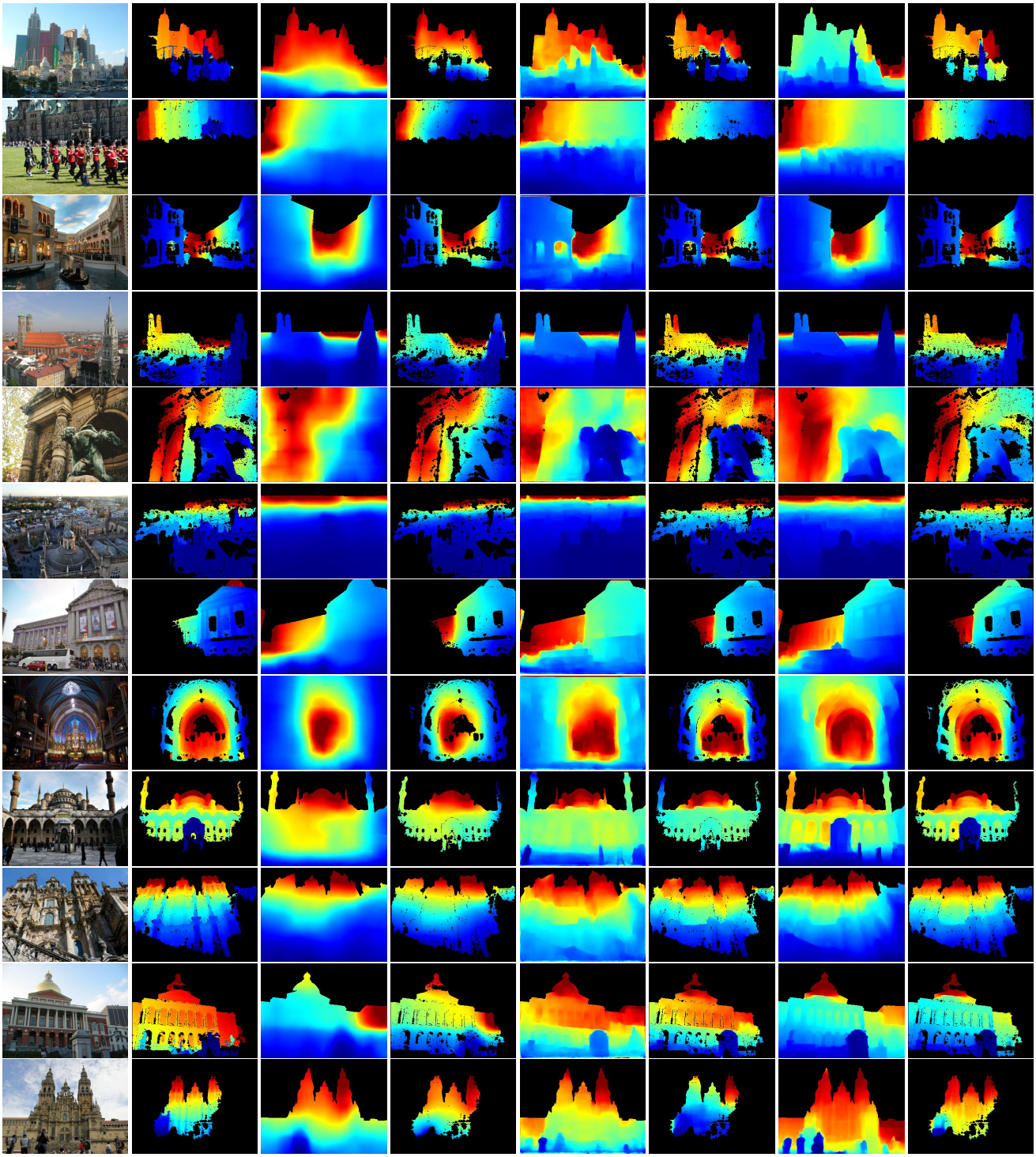
[3] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[4] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Int. Conf. on 3D Vision (3DV)*, pages 239–248, 2016.

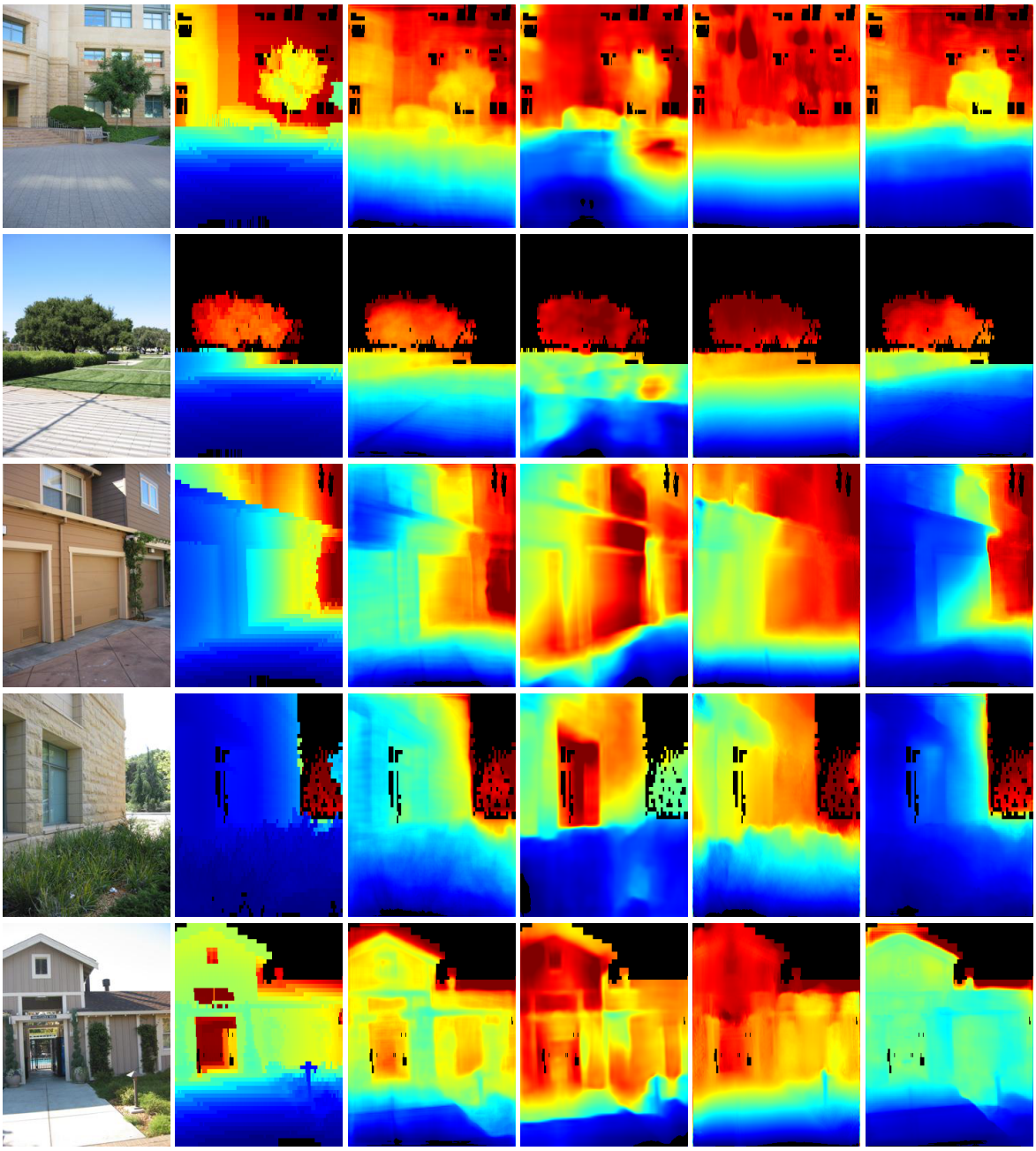
[6] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *Trans. Pattern Analysis and Machine Intelligence*, 2015.



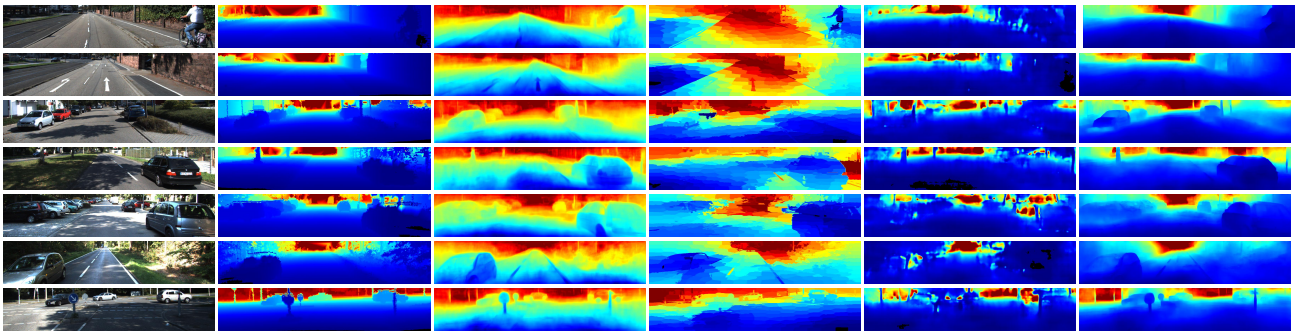
(a) Image (b) GT (c) VGG* (d) VGG* (M) (e) ResNet (f) ResNet (M) (g) HG (h) HG (M)

Figure 6: **Depth predictions on MD test set.** (Blue=near, red=far.) For visualization, we mask out the detected sky region. In the columns marked (M), we apply the mask from the GT depth map (indicating valid reconstructed depths) to the prediction map, to aid comparison with GT. (a) Input photo. (b) Ground truth COLMAP depth map (GT). VGG* prediction using the loss and network of [3]. (d) GT-masked version of (c). (e) Depth prediction from a ResNet [5]. (f) GT-masked version of (e). (g) Depth prediction from an hourglass (HG) network [2] . (h) GT-masked version of (g).



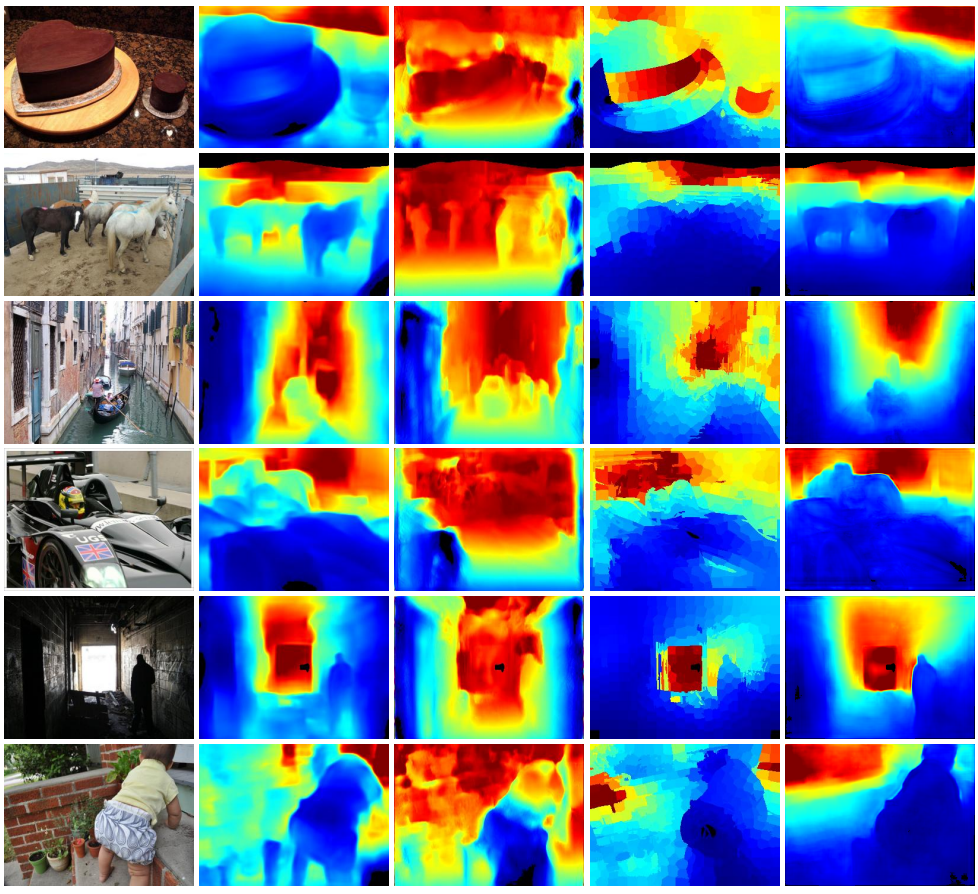
(a) Image (b) GT (c) DIW [2] (d) NYU [3] (e) KITTI [4] (f) Ours

Figure 7: **Depth predictions on Make3D.** (Blue=near, red=far.) (a) Input photo. (b) Ground truth (c) DIW-trained network predictions [2]. (d) Best NYU-trained network predictions [7] (e) Best KITTI-trained network predictions [4]. (f) Our MD-trained network predictions. None of the models were trained on Make3D data.



(a) Image (b) GT (c) DIW [2] (d) Best NYU [7] (e) Best Make3D [5] (f) Ours

Figure 8: **Depth predictions on KITTI.** (Blue=near, red=far.) (a) Input photo. (b) Ground truth (c) DIW-trained network predictions [2]. (d) Best NYU-trained network predictions [7] (e) Best Make3D-trained network predictions [5]. (f) Our MD-trained network predictions. None of the models were trained on KITTI data.



(a) Image (b) NYU [3] (c) KITTI [4] (d) Make3D [6] (e) Ours

Figure 9: **Depth predictions on the DIW test set.** (Blue=near, red=far.) (a) Input photo. (b) Best NYU-trained network predictions [3]. (c) Best KITTI-trained network predictions [4] (d) Best Make3D-trained network predictions [6]. (e) Our MD-trained network predictions. None of the models were trained on DIW data.