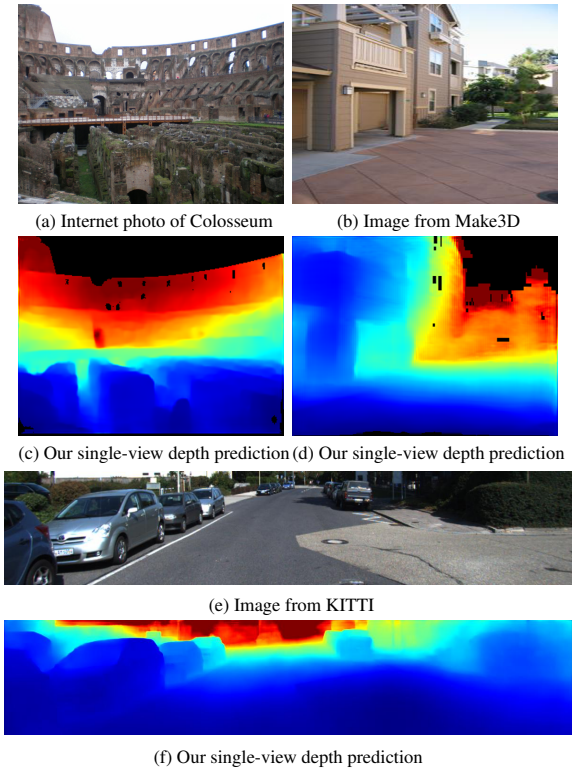
MegaDepth: Learning Single-View Depth Prediction from Internet Photos

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

Abstract

Single-view depth prediction is a fundamental problem in computer vision. Recently, deep learning methods have led to significant progress, but such methods are limited by the available training data. Current datasets based on 3D sensors have key limitations, including indoor-only images (NYU), small numbers of training examples (Make3D), and sparse sampling (KITTI). We propose to use multi-view Internet photo collections, a virtually unlimited data source, to generate training data via modern structure-from-motion and multi-view stereo (MVS) methods, and present a large depth dataset called MegaDepth based on this idea. Data derived from MVS comes with its own challenges, including noise and unreconstructable objects. We address these challenges with new data cleaning methods, as well as automatically augmenting our data with ordinal depth relations generated using semantic segmentation. We validate the use of large amounts of Internet data by showing that models trained on MegaDepth exhibit strong generalization—not only to novel scenes, but also to other diverse datasets including Make3D, KITTI, and DIW, even when no images from those datasets are seen during training.[1]



(a) Internet photo of Colosseum (b) Image from Make3D
(c) Our single-view depth prediction (d) Our single-view depth prediction
(e) Image from KITTI
(f) Our single-view depth prediction

Figure 1: We use large Internet image collections, combined with 3D reconstruction and semantic labeling methods, to generate large amounts of training data for single-view depth prediction. **(a), (b), (e):** Example input RGB images. **(c), (d), (f):** Depth maps predicted by our MegaDepth-trained CNN (blue=near, red=far). For these results, the network was not trained on Make3D and KITTI data.

1. Introduction

Predicting 3D shape from a single image is an important capability of visual reasoning, with applications in robotics, graphics, and other vision tasks such as intrinsic images. While single-view depth estimation is a challenging, underconstrained problem, deep learning methods have recently driven significant progress. Such methods thrive when trained with large amounts of data. Unfortunately, fully general training data in the form of (*RGB image, depth map*) pairs is difficult to collect. Commodity RGB-D sensors such as Kinect have been widely used for this purpose [34], but are limited to indoor use. Laser scanners have enabled important datasets such as Make3D [29] and KITTI [25], but such devices are cumbersome to operate (in the case of industrial scanners), or produce sparse depth maps (in the case of LIDAR). Moreover, both Make3D and KITTI are collected in specific scenarios (a university campus, and atop a car, respectively). Training data can also be generated through crowdsourcing, but this approach has so far been limited to gathering sparse ordinal relationships or surface normals [12, 4, 5].

In this paper, we explore the use of a nearly unlimited source of data for this problem: images from the Internet from overlapping viewpoints, from which structure-from-

[1]Project website: http://www.cs.cornell.edu/projects/megadepth/

motion (SfM) and multi-view stereo (MVS) methods can automatically produce dense depth. Such images have been widely used in research on large-scale 3D reconstruction [35, 14, 2, 8]. We propose to use the outputs of these systems as the inputs to machine learning methods for single-view depth prediction. By using large amounts of diverse training data from photos taken around the world, we seek to learn to predict depth with high accuracy and generalizability. Based on this idea, we introduce MegaDepth (MD), a large-scale depth dataset generated from Internet photo collections, which we make fully available to the community.

To our knowledge, ours is the first use of Internet SfM+MVS data for single-view depth prediction. Our main contribution is the MD dataset itself. In addition, in creating MD, we found that care must be taken in preparing a dataset from noisy MVS data, and so we also propose new methods for processing raw MVS output, and a corresponding new loss function for training models with this data. Notably, because MVS tends to not reconstruct dynamic objects (people, cars, etc), we augment our dataset with ordinal depth relationships automatically derived from semantic segmentation, and train with a joint loss that includes an ordinal term. In our experiments, we show that by training on MD, we can learn a model that works well not only on images of new scenes, but that also generalizes remarkably well to completely different datasets, including Make3D, KITTI, and DIW—achieving much better generalization than prior datasets. Figure 1 shows example results spanning different test sets from a network trained solely on our MD dataset.

2. Related work

Single-view depth prediction. A variety of methods have been proposed for single-view depth prediction, most recently by utilizing machine learning [15, 28]. A standard approach is to collect RGB images with ground truth depth, and then train a model (e.g., a CNN) to predict depth from RGB [7, 22, 23, 27, 3, 19]. Most such methods are trained on a few standard datasets, such as NYU [33, 34], Make3D [29], and KITTI [11], which are captured using RGB-D sensors (such as Kinect) or laser scanning. Such scanning methods have important limitations, as discussed in the introduction. Recently, Novotny *et al*. [26] trained a network on 3D models derived from SfM+MVS on videos to learn 3D shapes of single objects. However, their method is limited to images of objects, rather than scenes.

Multiple views of a scene can also be used as an implicit source of training data for single-view depth prediction, by utilizing view synthesis as a supervisory signal [38, 10, 13, 43]. However, view synthesis is only a proxy for depth, and may not always yield high-quality learned depth. Ummenhofer *et al*. [36] trained from overlapping image pairs taken with a single camera, and learned to predict image matches, camera poses, and depth. However, it requires two input images at test time.

Ordinal depth prediction. Another way to collect depth data for training is to ask people to manually annotate depth in images. While labeling absolute depth is challenging, people are good at specifying *relative* (ordinal) depth relationships (e.g., *closer-than*, *further-than*) [12]. Zoran *et al*. [44] used such relative depth judgments to predict ordinal relationships between points using CNNs. Chen *et al*. leveraged crowdsourcing of ordinal depth labels to create a large dataset called "Depth in the Wild" [4]. While useful for predicting depth ordering (and so we incorporate ordinal data automatically generated from our imagery), the Euclidean accuracy of depth learned solely from ordinal data is limited.

Depth estimation from Internet photos. Estimating geometry from Internet photo collections has been an active research area for a decade, with advances in both structure from motion [35, 2, 37, 30] and multi-view stereo [14, 9, 32]. These techniques generally operate on 10s to 1000s of images. Using such methods, past work has used retrieval and SfM to build a 3D model seeded from a single image [31], or registered a photo to an existing 3D model to transfer depth [40]. However, this work requires either having a detailed 3D model of each location in advance, or building one at run-time. Instead, we use SfM+MVS to train a network that generalizes to novel locations and scenarios.

3. The MegaDepth Dataset

In this section, we describe how we construct our dataset. We first download Internet photos from Flickr for a set of well-photographed landmarks from the Landmarks10K dataset [21]. We then reconstruct each landmark in 3D using state-of-the-art SfM and MVS methods. This yields an SfM model as well as a dense depth map for each reconstructed image. However, these depth maps have significant noise and outliers, and training a deep network on this raw depth data will not yield a useful predictor. Therefore, we propose a series of processing steps that prepare these depth maps for use in learning, and additionally use semantic segmentation to automatically generate ordinal depth data.

3.1. Photo calibration and reconstruction

We build a 3D model from each photo collection using COLMAP, a state-of-art SfM system [30] (for reconstructing camera poses and sparse point clouds) and MVS system [32] (for generating dense depth maps). We use COLMAP because we found that it produces high-quality 3D models via its careful incremental SfM procedure, but other such systems could be used. COLMAP produces a depth map D for every reconstructed photo I (where some pixels of D can be empty if COLMAP was unable to recover a depth), as well as other outputs, such as camera parameters and sparse SfM points plus camera visibility.

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

Figure 2: **Comparison between MVS depth maps with and without our proposed refinement/cleaning methods.** The raw MVS depth maps (middle) exhibit depth bleeding (top) or incorrect depth on people (bottom). Our methods (right) can correct or remove such outlier depths.

3.2. Depth map refinement

The raw depth maps from COLMAP contain many outliers from a range of sources, including: (1) transient objects (people, cars, etc.) that appear in a single image but nonetheless are assigned (incorrect) depths, (2) noisy depth discontinuities, and (3) bleeding of background depths into foreground objects. Other MVS methods exhibit similar problems due to inherent ambiguities in stereo matching. Figure 2(b) shows two example depth maps produced by COLMAP that illustrate these issues. Such outliers have a highly negative effect on the depth prediction networks we seek to train. To address this problem, we propose two new depth refinement methods designed to generate high-quality training data:

First, we devise a modified MVS algorithm based on COLMAP, but more conservative in its depth estimates, based on the idea that we would prefer less training data over bad training data. COLMAP computes depth maps iteratively, at each stage trying to ensure geometric consistency between nearby depth maps. One adverse effect of this strategy is that background depths can tend to "eat away" at foreground objects, because one way to increase consistency between depth maps is to consistently predict the background depth (see Figure 2 (top)). To counter this effect, at each depth inference iteration in COLMAP, we compare the depth values at each pixel before and after the update and keep the smaller (closer) of the two. We then apply a median filter to remove unstable depth values. We describe our modified MVS algorithm in detail in the supplemental material.

Second, we utilize semantic segmentation to enhance and filter the depth maps, and to yield large amounts of ordinal depth comparisons as additional training data. The second row of Figure 2 shows an example depth map computed with our object-aware filtering. We now describe our use of semantic segmentation in detail.

3.3. Depth enhancement via semantic segmentation

Multi-view stereo methods can have problems with a number of object types, including transient objects such as people and cars, difficult-to-reconstruct objects such as poles and traffic signals, and sky regions. However, if we can understand the semantic layout of an image, then we can attempt to mitigate these issues, or at least identify problematic pixels. We have found that deep learning methods for semantic segmentation are starting to become reliable enough for this use [41].

We propose three new uses of semantic segmentation in the creation of our dataset. First, we use such segmentations to remove spurious MVS depths in foreground regions. Second, we use the segmentation as a criterion to categorize each photo as providing either Euclidean depth or ordinal depth data. Finally, we combine semantic information and MVS depth to automatically annotate ordinal depth relationships, which can be used to help training in regions that cannot be reconstructed by MVS.

Semantic filtering. To process a given photo I, we first run semantic segmentation using PSPNet [41], a recent segmentation method, trained on the MIT Scene Parsing dataset (consisting of 150 semantic categories) [42]. We then divide the pixels into three subsets by predicted semantic category:

1. **Foreground objects**, denoted F, corresponding to objects that often appear in the foreground of scenes, including static foreground objects (e.g., statues, fountains) and dynamic objects (e.g., people, cars).
2. **Background objects**, denoted B, including buildings, towers, mountains, etc. (See supplemental material for full details of the foreground/background classes.)
3. **Sky**, denoted S, which is treated as a special case in the depth filtering described below.

We use this semantic categorization of pixels in several ways. As illustrated in Figure 2 (bottom), transient objects such as people can result in spurious depths. To remove these from each image I, we consider each connected component C of the foreground mask F. If $< 50\%$ of pixels in C have a reconstructed depth, we discard all depths from C. We use a threshold of 50%, rather than simply removing all foreground depths, because pixels on certain objects in F (such as sculptures) can indeed be accurately reconstructed (and we found that PSPNet can sometimes mistake sculptures and people for one another). This simple filtering of foreground depths yields large improvements in depth map quality. Additionally, we remove reconstructed depths that fall inside the sky region S, as such depths tend to be spurious.

Euclidean vs. ordinal depth. For each 3D model we have thousands of reconstructed Internet photos, and ideally we would use as much of this depth data as possible for training. However, some depth maps are more reliable than others, due

Figure 3: **Examples of automatic ordinal labeling.** Blue mask: foreground (F_{ord}) derived from semantic segmentation. Red mask: background (B_{ord}) derived from reconstructed depth.

to factors such as the accuracy of the estimated camera pose or the presence of large occluders. Hence, we found that it is beneficial to limit training to a subset of highly reliable depth maps. We devise a simple but effective way to compute a subset of high-quality depth maps, by thresholding by the fraction of reconstructed pixels. In particular, if $\geq 30\%$ of an image I (ignoring the sky region S) consists of valid depth values, then we keep that image as training data for learning Euclidean depth. This criterion prefers images without large transient foreground objects (e.g., "no selfies"). At the same time, such foreground-heavy images are extremely useful for another purpose: automatically generating training data for learning *ordinal* depth relationships.

Automatic ordinal depth labeling. As noted above, transient or difficult to reconstruct objects, such as people, cars, and street signs are often missing from MVS reconstructions. Therefore, using Internet-derived data alone, we will lack ground truth depth for such objects, and will likely do a poor job of learning to reconstruct them. To address this issue, we propose a novel method of automatically extracting ordinal depth labels from our training images based on their estimated 3D geometry and semantic segmentation.

Let us denote as O ("Ordinal") the subset of photos that do *not* satisfy the "no selfies" criterion described above. For each image $I \in O$, we compute two regions, $F_{\text{ord}} \in F$ (based on semantic information) and $B_{\text{ord}} \in B$ (based on 3D geometry information), such that all pixels in F_{ord} are likely closer to the camera than all pixels in B_{ord}. Briefly, F_{ord} consists of large connected components of F, and B_{ord} consists of large components of B that also contain valid depths in the last quartile of the full depth range for I (see supplementary for full details). We found this simple approach works very well ($> 95\%$ accuracy in pairwise ordinal relationships), likely because natural photos tend to be composed in certain common ways. Several examples of our automatic ordinal depth labels are shown in Figure 3.

3.4. Creating a dataset

We use the approach above to densely reconstruct 200 3D models from landmarks around the world, representing about 150K reconstructed images. After our proposed filtering, we are left with 130K valid images. Of these 130K photos, around 100K images are used for Euclidean depth data, and the remaining 30K images are used to derive ordinal depth data. We also include images from [18] in our training set. Together, this data comprises the MegaDepth (MD) dataset, available at http://www.cs.cornell.edu/projects/megadepth/.

4. Depth estimation network

This section presents our end-to-end deep learning algorithm for predicting depth from a single photo.

4.1. Network architecture

We evaluated three networks used in prior work on single-view depth prediction: VGG [6], the "hourglass" network [4], and a ResNet architecture [19]. Of these, the hourglass network performed best, as described in Section 5.

4.2. Loss function

The 3D data produced by SfM+MVS is only up to an unknown scale factor, so we cannot compare predicted and ground truth depths directly. However, as noted by Eigen and Fergus [7], the *ratios of pairs of depths* are preserved under scaling (or, in the log-depth domain, the difference between pairs of log-depths). Therefore, we solve for a depth map in the log domain and train using a scale-invariant loss function, $\mathcal{L}_{\text{si}}$. $\mathcal{L}_{\text{si}}$ combines three terms:

$$\mathcal{L}_{\text{si}} = \mathcal{L}_{\text{data}} + \alpha \mathcal{L}_{\text{grad}} + \beta \mathcal{L}_{\text{ord}}. \tag{1}$$

Scale-invariant data term. We adopt the loss of Eigen and Fergus [7], which computes the mean square error (MSE) of the difference between *all* pairs of log-depths in linear time. Suppose we have a predicted log-depth map L, and a ground truth log depth map L^*. L_i and L_i^* denote corresponding individual log-depth values indexed by pixel position i. We denote $R_i = L_i - L_i^*$ and define:

$$\mathcal{L}_{\text{data}} = \frac{1}{n}\sum_{i=1}^{n}(R_i)^2 - \frac{1}{n^2}\left(\sum_{i=1}^{n} R_i\right)^2 \tag{2}$$

where n is the number of valid depths in the ground truth depth map.

Multi-scale scale-invariant gradient matching term. To encourage smoother gradient changes and sharper depth discontinuities in the predicted depth map, we introduce a multi-scale scale-invariant gradient matching term $\mathcal{L}_{\text{grad}}$, defined as an ℓ_1 penalty on differences in log-depth gradients between the predicted and ground truth depth map:

$$\mathcal{L}_{\text{grad}} = \frac{1}{n}\sum_{k}\sum_{i}\left(\left|\nabla_x R_i^k\right| + \left|\nabla_y R_i^k\right|\right) \tag{3}$$

where R_i^k is the value of the log-depth difference map at position i and scale k. Because the loss is computed at

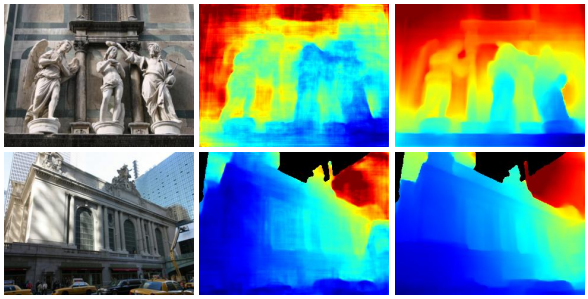


Figure 4: **Effect of $\mathcal{L}_{\text{grad}}$ term.** $\mathcal{L}_{\text{grad}}$ encourages predictions to match the ground truth depth gradient.

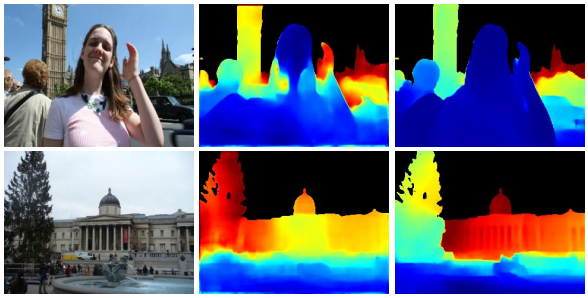


Figure 5: **Effect of $\mathcal{L}_{\text{ord}}$ term.** $\mathcal{L}_{\text{ord}}$ tends to corrects ordinal depth relations for hard-to-construct objects such as the person in the first row and the tree in the second row.

multiple scales, $\mathcal{L}_{\text{grad}}$ captures depth gradients across large image distances. In our experiments, we use four scales. We illustrate the effect of $\mathcal{L}_{\text{grad}}$ in Figure 4.

Robust ordinal depth loss. Inspired by Chen *et al*. [4], our ordinal depth loss term $\mathcal{L}_{\text{ord}}$ utilizes the automatic ordinal relations described in Section 3.3. During training, for each image in our ordinal set O, we pick a single pair of pixels (i, j), with pixel i and j either belonging to the foreground region F_{ord} or the background region B_{ord}. $\mathcal{L}_{\text{ord}}$ is designed to be robust to the small number of incorrectly ordered pairs.

$$\mathcal{L}_{\text{ord}} = \begin{cases} \log\left(1 + \exp\left(P_{ij}\right)\right) & \text{if } P_{ij} \leq \tau \\ \log\left(1 + \exp\left(\sqrt{P_{ij}}\right)\right) + c & \text{if } P_{ij} > \tau \end{cases} \quad (4)$$

where $P_{ij} = -r^*_{ij}\left(L_i - L_j\right)$ and r^*_{ij} is the automatically labeled ordinal depth relation between i and j ($r^*_{ij} = 1$ if pixel i is further than j and -1 otherwise). c is a constant set so that $\mathcal{L}_{\text{ord}}$ is continuous. $\mathcal{L}_{\text{ord}}$ encourages the depth difference of a pair of points to be large (and ordered) if our automatic labeling method judged the pair to have a likely depth ordering. We illustrate the effect of $\mathcal{L}_{\text{ord}}$ in Figure 5. In our tests, we set $\tau = 0.25$ based on cross-validation.

5. Evaluation

In this section, we evaluate our networks on a number of datasets, and compare to several state-of-art depth prediction algorithms, trained on a variety of training data. In our evaluation, we seek to answer several questions, including:

- How well does our model trained on MD generalize to new Internet photos from never-before-seen locations?
- How important is our depth map processing? What is the effect of the terms in our loss function?
- How well does our model trained on MD generalize to other types of images from other datasets?

The third question is perhaps the most interesting, because the promise of training on large amounts of diverse data is good generalization. Therefore, we run a set of experiments training on one dataset and testing on another, and show that our MD dataset gives the best generalization performance.

We also show that our depth refinement strategies are essential for achieving good generalization, and show that our proposed loss function—combining scale-invariant data terms with an ordinal depth loss—improves prediction performance both quantitatively and qualitatively.

Experimental setup. Out of the 200 reconstructed models in our MD dataset, we randomly select 46 to form a test set (locations not seen during training). For the remaining 154 models, we randomly split images from each model into training and validation sets with a ratio of 96% and 4% respectively. We set $\alpha = 0.5$ and $\beta = 0.1$ using MD validation set. We implement our networks in PyTorch [1], and train using Adam [17] for 20 epochs with batch size 32.

For fair comparison, we train and validate our network using MD data for all experiments. Due to variance in performance of cross-dataset testing, we train four models on MD and compute the average error (see supplemental material for the performance of each individual model).

5.1. Evaluation and ablation study on MD test set

In this subsection, we describe experiments where we train on our MD training set and test on the MD test set.

Error metrics. For numerical evaluation, we use two scale-invariant error measures (as with our loss function, we use scale-invariant measures due to the scale-free nature of SfM models). The first measure is the scale-invariant RMSE (si-RMSE) (Equation 2), which measures precise numerical depth accuracy. The second measure is based on the preservation of depth ordering. In particular, we use a measure similar to [44, 4] that we call the *SfM Disagreement Rate* (SDR). SDR is based on the rate of disagreement with ordinal depth relationships derived from estimated SfM points. We use sparse SfM points rather than dense MVS because we found that sparse SfM points capture some structures not

Network	si-RMSE	SDR$^{=}$%	SDR$^{\neq}$%	SDR%
VGG* [6]	0.116	31.28	28.63	29.78
VGG (full)	0.115	29.64	27.22	28.40
ResNet (full)	0.124	**27.32**	25.35	26.27
HG (full)	**0.104**	27.73	**24.36**	**25.82**

Table 1: **Results on the MD test set (places unseen during training) for several network architectures.** For VGG* we use the same loss and network architecture as in [6] for comparison to [6]. Lower is better.

Method	si-RMSE	SDR$^{=}$%	SDR$^{\neq}$%	SDR%
$\mathcal{L}_{\text{data}}$ only	0.148	33.20	30.65	31.75
+$\mathcal{L}_{\text{grad}}$	0.123	**26.17**	28.32	27.11
+$\mathcal{L}_{\text{grad}}$ +$\mathcal{L}_{\text{ord}}$	**0.104**	27.73	**24.36**	**25.82**

Table 2: **Results on MD test set (places unseen during training) for different loss configurations.** Lower is better.

reconstructed by MVS (e.g., complex objects such as lamp-posts). We define SDR(D, D^{*}), the ordinal disagreement rate between the predicted (non-log) depth map $D = \exp(L)$ and ground-truth SfM depths D^{*}, as:

$$\text{SDR}(D, D^{*}) = \frac{1}{n} \sum_{i,j \in \mathcal{P}} \mathbb{1}\left(\text{ord}(D_i, D_j) \neq \text{ord}(D_i^{*}, D_j^{*})\right) \quad (5)$$

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

$$\text{ord}(D_i, D_j) = \begin{cases} 1 & \text{if } \frac{D_i}{D_j} > 1 + \delta \\ -1 & \text{if } \frac{D_i}{D_j} < 1 - \delta \\ 0 & \text{if } 1 - \delta \leq \frac{D_i}{D_j} \leq 1 + \delta \end{cases} \quad (6)$$

We also define SDR$^{=}$ and 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. For efficiency, we sample SfM points from the full set to compute this error term.

Effect of network and loss variants. We evaluate three popular network architectures for depth prediction on our MD test set: the VGG network used by Eigen *et al*. [6], an "hourglass"(HG) network [4], and ResNets [19]. To compare our loss function to that of Eigen *et al*. [6], we also test the same network and loss function as [6] trained on MD. [6] uses a VGG network with a scale-invariant loss plus single scale gradient matching term. Quantitative results are shown in Table 1 and qualitative comparisons are shown in Figure 6. We also evaluate variants of our method trained using only some of our loss terms: (1) a version with only the scale-invariant data term $\mathcal{L}_{\text{data}}$ (the same loss as in [7]), (2) a version that adds our multi-scale gradient matching loss $\mathcal{L}_{\text{grad}}$, and (3) the full version including $\mathcal{L}_{\text{grad}}$ and the ordinal depth loss $\mathcal{L}_{\text{ord}}$. Results are shown in Table 2.

Test set	Error measure	Raw MD	Clean MD
Make3D	RMS	11.41	**5.322**
	Abs Rel	0.614	**0.364**
	log10	0.386	**0.152**
KITTI	RMS	12.15	**6.680**
	RMS(log)	0.582	**0.414**
	Abs Rel	0.433	**0.368**
	Sq Rel	3.927	**2.587**
DIW	WHDR%	31.32	**24.55**

Table 3: **Results on three different test sets with and without our depth refinement methods**. *Raw MD* indicates raw depth data; *Clean MD* indicates depth data using our refinement methods. Lower is better for all error measures.

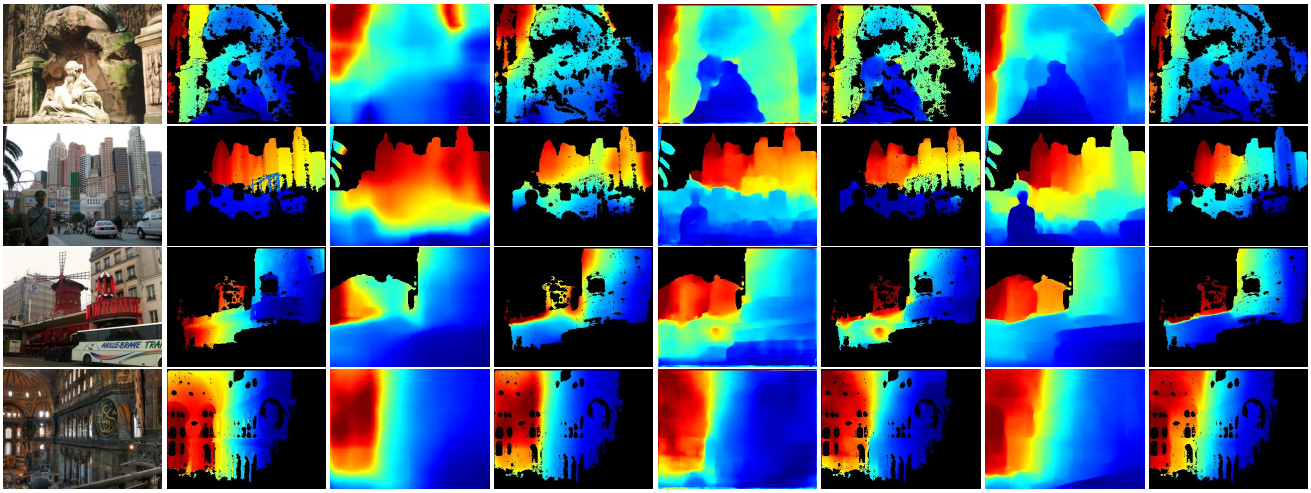
As shown in Tables 1 and 2, the HG architecture achieves the best performance of the three architectures, and training with our full loss yields better performance compared to other loss variants, including that of [6] (first row of Table 1). One thing to notice that is adding $\mathcal{L}_{\text{ord}}$ could significantly improve SDR$^{\neq}$ while increasing SDR$^{=}$. Figure 6 shows that our joint loss helps preserve the structure of the depth map and capture nearby objects such as people and buses.

Finally, we experiment with training our network on MD with and without our proposed depth refinement methods, testing on three datasets: KITTI, Make3D, and DIW. The results, shown in Table 3, show that networks trained on raw MVS depth do not generalize well. Our proposed refinements significantly boost prediction performance.

5.2. Generalization to other datasets

A powerful application of our 3D-reconstruction-derived training data is to generalize to outdoor images beyond landmark photos. To evaluate this capability, we train our model on MD and test on three standard benchmarks: Make3D [28], KITTI [11], and DIW [4]—*without* seeing training data from these datasets. Since our depth prediction is defined up to a scale factor, for each dataset, we align each prediction from all non-target dataset trained models with the ground truth by a scalar computed from least sqaure solution to the ratio between ground truth and predicted depth.

Make3D. To test on Make3D, we follow the protocol of [23, 19], ,resizing all images to 345×460, and removing ground truth depths larger than 70m (since Make3D data is unreliable at large distances). We train our network only on MD using our full loss. Table 4 shows numerical results, including comparisons to several methods trained on both Make3D and non-Make3D data, and Figure 7 visualizes



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

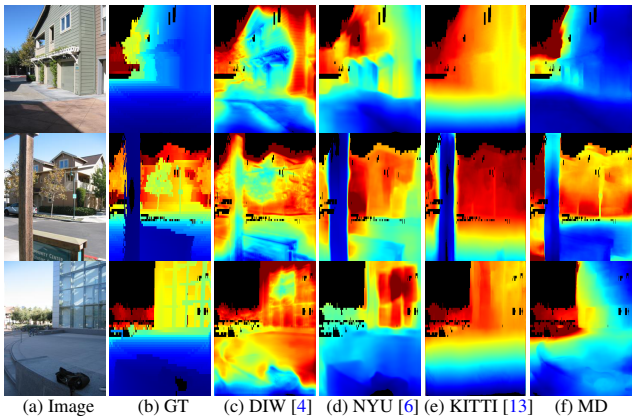
Figure 6: **Depth predictions on MD test set.** (Blue=near, red=far.) For visualization, we mask out the detected sky region. In the columns marked (M), we apply the mask from the GT depth map (indicating valid reconstructed depths) to the prediction map, to aid comparison with GT. (a) Input photo. (b) Ground truth COLMAP depth map (GT). VGG* prediction using the loss and network of [6]. (d) GT-masked version of (c). (e) Depth prediction from a ResNet [19]. (f) GT-masked version of (e). (g) Depth prediction from an hourglass (HG) network [4] . (h) GT-masked version of (g).

Training set	Method	RMS	Abs Rel	log10
Make3D	Karsch *et al.* [16]	9.20	0.355	0.127
	Liu *et al.* [24]	9.49	0.335	0.137
	Liu *et al.* [22]	8.60	0.314	0.119
	Li *et al.* [20]	7.19	0.278	0.092
	Laina *et al.* [19]	4.45	0.176	0.072
	Xu *et al.* [39]	4.38	0.184	0.065
NYU	Eigen *et al.* [6]	6.89	0.505	0.198
	Liu *et al.* [22]	7.20	0.669	0.212
	Laina *et al.* [19]	7.31	0.669	0.216
KITTI	Zhou *et al.* [43]	8.39	0.651	0.231
	Godard *et al.* [13]	9.88	0.525	0.319
DIW	Chen *et al.* [4]	7.25	0.550	0.200
MD	Ours	6.23	0.402	0.156
MD+Make3D	Ours	4.25	0.178	0.064

Table 4: **Results on Make3D for various training datasets and methods.** The first column indicates the training dataset. Errors for "Ours" are averaged over four models trained/validated on MD. Lower is better for all metrics.

depth predictions from our model and several other non-Make3D-trained models. Our network trained on MD have the best performance among all non-Make3D-trained models. Finally, the last row of Table 4 shows that our model fine-tuned on Make3D achieves better performance than the state-of-the-art.

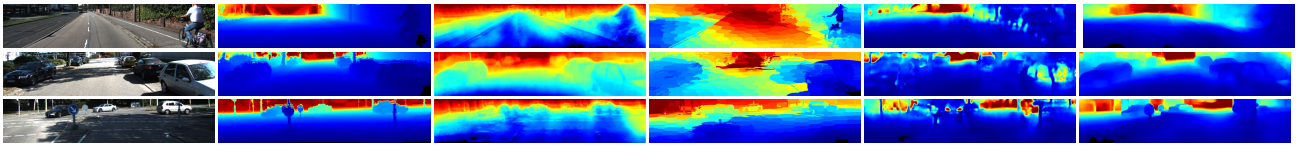
KITTI. Next, we evaluate our model on the KITTI test set



(a) Image (b) GT (c) DIW [4] (d) NYU [6] (e) KITTI [13] (f) MD

Figure 7: **Depth predictions on Make3D.** The last four columns show results from the best models trained on non-Make3D datasets (final column is our result).

based on the split of [7]. As with our Make3D experiments, we do not use images from KITTI during training. The KITTI dataset is very different from ours, consisting of driving sequences that include objects, such as sidewalks, cars, and people, that are difficult to reconstruct with SfM/MVS. Nevertheless, as shown in Table 5, our MD-trained network still outperform approaches trained on non-KITTI datasets. Finally, the last row of Table 5 shows that we can achieve state-of-the-art performance by fine-tuning our network on KITTI training data. Figure 8 shows visual comparisons between our results and models trained on other non-KITTI



(a) Image (b) GT (c) DIW [4] (d) Best NYU [23] (e) Best Make3D [19] (f) MD

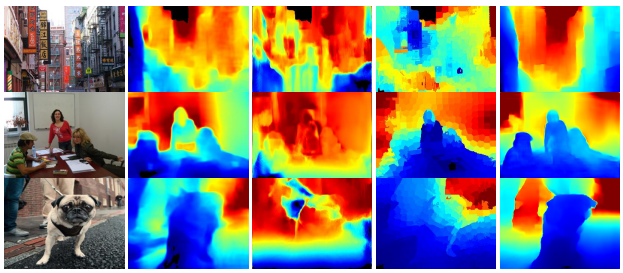
Figure 8: **Depth predictions on KITTI.** (Blue=near, red=far.) None of the models were trained on KITTI data.

Training set	Method	RMS	RMS(log)	Abs Rel	Sq Rel
KITTI	Liu *et al.* [23]	6.52	0.275	0.202	1.614
	Eigen *et al.* [7]	6.31	0.282	0.203	1.548
	Zhou *et al.* [43]	6.86	0.283	0.208	1.768
	Godard *et al.* [13]	5.93	0.247	0.148	1.334
Make3D	Laina *et al.* [19]	8.68	0.422	0.339	3.136
	Liu *et al.* [22]	8.70	0.447	0.362	3.465
NYU	Eigen *et al.* [6]	10.37	0.510	0.521	5.016
	Liu *et al.* [22]	10.10	0.526	0.540	5.059
	Laina *et al.* [19]	10.07	0.527	0.515	5.049
CS	Zhou *et al.* [43]	7.58	0.334	0.267	2.686
DIW	Chen *et al.* [4]	7.12	0.474	0.393	3.260
MD	Ours	6.68	0.414	0.368	2.587
MD+KITTI	Ours	5.25	0.229	0.139	1.325

Table 5: **Results on the KITTI test set for various training datasets and approaches.** Columns are as in Table 4.

Training set	Method	WHDR%
DIW	Chen *et al.* [4]	22.14
KITTI	Zhou *et al.* [43]	31.24
	Godard *et al.* [13]	30.52
NYU	Eigen *et al.* [6]	25.70
	Laina *et al.* [19]	45.30
	Liu *et al.* [22]	28.27
Make3D	Laina *et al.* [19]	31.65
	Liu *et al.* [22]	29.58
MD	Ours	24.55

Table 6: **Results on the DIW test set for various training datasets and approaches.** Columns are as in Table 4.



(a) Image (b) NYU [6] (c) KITTI [13] (d) Make3D [22] (e) Ours

Figure 9: **Depth predictions on the DIW test set.** (Blue=near, red=far.) Captions are described in Figure 8. None of the models were trained on DIW data.

datasets. One can see that we achieve much better visual quality compared to other non-KITTI datasets, and our predictions can reasonably capture nearby objects such as traffic signs, cars, and trees, due to our ordinal depth loss.

DIW. Finally, we test our network on the DIW dataset [4]. DIW consists of Internet photos with general scene structures. Each image in DIW has a single pair of points with a human-labeled ordinal depth relationship. As with Make3D and KITTI, we do not use DIW data during training. For DIW, quality is computed via the *Weighted Human Disagreement Rate* (WHDR), which measures the frequency of disagreement between predicted depth maps and human annotations on a test set. Numerical results are shown in Table 6. Our MD-trained network again has the best performance among all non-DIW trained models. Figure 9 visualizes our predictions and those of other non-DIW-trained networks on DIW test images. Our predictions achieve visually better depth relationships. Our method even works reasonably well for challenging scenes such as offices and close-ups.

6. Conclusion

We presented a new use for Internet-derived SfM+MVS data: generating large amounts of training data for single-view depth prediction. We demonstrated that this data can be used to predict state-of-the-art depth maps for locations never observed during training, and generalizes very well to other datasets. However, our method also has a number of limitations. MVS methods still do not perfectly reconstruct even static scenes, particularly when there are oblique surfaces (e.g., ground), thin or complex objects (e.g., lamp-posts), and difficult materials (e.g., shiny glass). Our method does not predict metric depth; future work in SfM could use learning or semantic information to correctly scale scenes. Our dataset is currently biased towards outdoor landmarks, though by scaling to much larger input photo collections we will find more diverse scenes. Despite these limitations, our work points towards the Internet as an intriguing, useful source of data for geometric learning problems.

Acknowledgments. We thank the anonymous reviewers for their valuable comments. This work was funded by the National Science Foundation under grant IIS-1149393.

References

[1] Pytorch. 2016. http://pytorch.org.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2009.

[3] M. H. Baig and L. Torresani. Coupled depth learning. In *Proc. Winter Conf. on Computer Vision (WACV)*, 2016.

[4] 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.

[5] W. Chen, D. Xiang, and J. Deng. Surface normals in the wild. *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 1557–1566, 2017.

[6] 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.

[7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems*, pages 2366–2374, 2014.

[8] J.-M. Frahm, P. F. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building Rome on a cloudless day. In *Proc. European Conf. on Computer Vision (ECCV)*, 2010.

[9] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 1434–1441, 2010.

[10] R. Garg, G. Carneiro, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 740–756, 2016.

[11] A. Geiger. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] Y. I. Gingold, A. Shamir, and D. Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Trans. Graphics*, 2012.

[13] 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.

[14] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

[15] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 654–661, 2005.

[16] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 775–788, 2012.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graphics*, 36(4), 2017.

[19] 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.

[20] 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. Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, 2015.

[21] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *Large-Scale Visual Geo-Localization*, pages 147–163. Springer, 2016.

[22] 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.

[23] 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*, 38:2024–2039, 2016.

[24] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, 2014.

[25] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] D. Novotny, D. Larlus, and A. Vedaldi. Learning 3d object categories by looking around them. *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 5218–5227, 2017.

[27] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, volume 18, pages 1–8, 2005.

[29] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *Trans. Pattern Analysis and Machine Intelligence*, 31(5), 2009.

[30] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[31] J. L. Schönberger, F. Radenovic, O. Chum, and J.-M. Frahm. From single image query to detailed 3D reconstruction. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[32] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 501–518, 2016.

[33] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV Workshops*, 2011.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 746–760, 2012.

[35] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM Trans. Graphics (SIGGRAPH)*, 2006.

[36] 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. Computer Vision and Pattern Recognition (CVPR)*, pages 5622–5631, 2017.

[37] C. Wu. Towards linear-time incremental structure from motion. In *Int. Conf. on 3D Vision (3DV)*, 2013.

[38] J. Xie, R. B. Girshick, and A. Farhadi. Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016.

[39] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] C. Zhang, J. Gao, O. Wang, P. F. Georgel, R. Yang, J. Davis, J.-M. Frahm, and M. Pollefeys. Personal photograph enhancement using internet photo collections. *IEEE Trans. Visualization and Computer Graphics*, 2014.

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[44] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 388–396, 2015.