Learning Single-Image Depth from Videos using Quality Assessment Networks

Weifeng Chen, Jia Deng

University of Michigan, Ann Arbor
{wfchen, jiadeng}@umich.edu

Abstract. Although significant progress has been made in recent years, depth estimation from a single image in the wild is still a very challenging problem. One reason is the lack of high-quality image-depth data in the wild. In this paper we propose a fully automatic pipeline based on Structure-from-Motion (SfM) to generate such data from arbitrary videos. The core of this pipeline is a Quality Assessment Network that can distinguish correct and incorrect reconstructions obtained from SfM. With the proposed pipeline, we generate image-depth data from the NYU Depth dataset and random YouTube videos. We show that depth-prediction networks trained on such data can achieve competitive performance on the NYU Depth and the Depth-in-the-Wild benchmarks.

Keywords: Depth Recovery, Structure-from-Motion

1 Introduction

Estimating depth from a single image is an important problem in vision. Thanks to the availability of large image-depth datasets [1, 2], data-driven approaches [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19] can now produce high-quality depth predictions on various benchmarks. However, it remains a challenge to extend such success to predicting depth from images in the wild, i.e., images that depict a variety of scenes and content [20]. One major hindrance is the lack of diverse training data. Most available datasets depict either indoor [2, 21, 22] or road scenes [1], because they are collected via depth-sensing devices that work only under limited conditions. Chen et al. [20] tackle this problem by crowdsourcing the data collection task to collect relative-depth annotations from Internet images. Their Depth-in-the-Wild (DIW) dataset captures a broad range of content, but requires a significant amount of manual labor to collect. Using computer graphics to render image-depth data [23, 24, 25] is another solution. However, it remains unclear how to automatically generate data that can match the diversity of real-world images.

In this paper, we propose to run Structure-from-Motion (SfM) on Internet videos to obtain a large quantity of in-the-wild image-depth data. For each video, the SfM system reconstructs a 3D point cloud, which is then projected to depth values in each image to obtain image-depth data. This is motivated by two

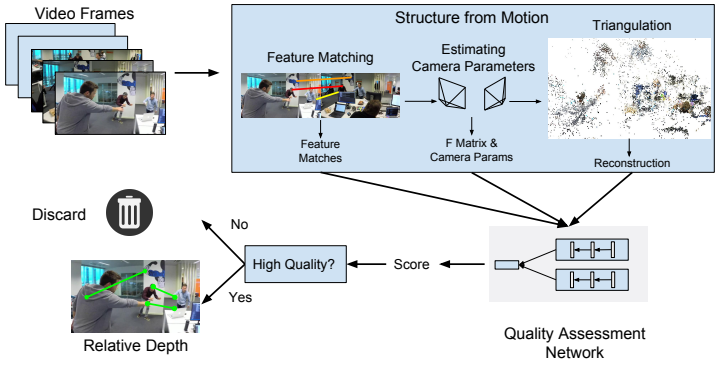

Fig. 1. An overview of our data collection pipeline. Given an arbitrary video, we follow standard steps of structure-from-motion: extracting feature points and matching them across frames, estimating the camera parameters, and performing triangulation to obtain a reconstruction. The reconstruction is then fed into the Quality Assessment Network (QANet) along with intermediate outputs of the SfM system, including feature matches, the fundamental matrix and the recovered camera parameters. The reconstruction is then assigned a score. If the score is above a certain threshold, this reconstruction is deemed of high quality, and we move on to extract relative depth annotations from it. Otherwise, the reconstruction is discarded

observations. First, SfM is a well-studied technique that enables us to reconstruct 3D structures from videos or image collections, and has been applied to problems in the wild with success [26, 27, 28, 29, 30]. Second, millions of videos are readily available on the Internet providing a wide array of diverse content. It is promising to leverage these two facts to obtain a large amount of image-depth data.

However, in practice significant technical difficulties arise when applying an off-the-shelf SfM system to arbitrary Internet videos. Typically, a SfM system consists of several stages: feature extraction and matching, triangulation, and bundle-adjustment. Failures at each stage are common and can be caused by a number of factors, including: feature mismatches, unknown camera intrinsics, and the presence of moving objects. Such failures result in erroneous reconstructions, and this raises concerns if we wish to obtain high-quality image-depth data.

In this paper, we propose a *Quality Assessment Network* (*QANet*) to determine the quality of a reconstruction. The network estimates a quality score by examining various outputs of an SfM pipeline, including the final 3D point cloud as well as intermediate outputs such as feature matches, camera parameters, and the fundamental matrix. By integrating the QANet into a SfM system, we propose a data-collection pipeline that enables us to obtain high-quality image-depth data in a fully automatic manner.

We experiment with this data-collection pipeline on various image sequences. Results on the NYU Depth dataset [2] show that this pipeline can produce high-quality image-depth data to train better depth-prediction networks [20]. We then

apply the proposed pipeline on 900,000 YouTube videos. The resulting dataset is called *YouTube3D*[1]. We train depth-prediction networks on a combination of YouTube3D and various amounts of the human annotated DIW dataset [20]. The test results on the DIW test set show that networks trained purely on the auto-generated YouTube3D can already achieve competitive performance. Moreover, when human annotations are provided, YouTube3D is a valuable supplement that provides extra training supervision to help obtain superior depth predictions in the wild.

To summarize, our contributions are as follows:

1. We propose a *Quality Assessment Network* (*QANet*) to distinguish high-quality SfM reconstructions from unsatisfactory ones. With this network, we propose a fully automatic and scalable pipeline to produce a large amount of high-quality image-depth data from Internet videos.

2. We contribute a large-scale image-depth dataset in the wild (*YouTube3D*). By leveraging YouTube3D and human annotated data [20] we obtain state-of-the-art performance on depth prediction in the wild.

2 Related Work

2.1 RGB-Depth Datasets

The key to the recent success of image-to-depth methods is the availability of large-scale RGB-D datasets [1, 2]. However, these datasets not only require a great deal of manual labor to collect, but also have limited scene diversity. For example, KITTI [1] consists mainly of road scenes; ETH3D [31] depicts a limited number of indoor and outdoor scenes; NYU Depth [2], ScanNet [21] and Matterport3D [22] only consist of indoor scenes. This lack of diversity makes it hard for depth-prediction methods trained on these datasets to generalize well in the wild.

Rendering ground-truth RGB-D data with computer graphics techniques is another option. Mayer et al. [24] obtain high-quality disparity and optical flow maps by training networks on synthetic data. Tatarchenko et al. [32] train networks on synthetic car images to estimate multi-view 3D models from single images. Synthetic RGB-D datasets such as SUNCG [33], SceneNet [34] and MPI-Sintel [23] have also been utilized in numerous tasks with success. However, these rendered data are either unrealistic [24, 23] or lacking in diversity [33, 34]. Although there is a study on the effects of rendering methods on training vision tasks [35], it remains unclear how to render RGB-D data that can capture the variety and properties of real-world scenes.

Our proposed data-collection pipeline addresses these issues by directly collecting image-depth ground-truth from Internet videos. Moreover, the data collection process is fully automatic and requires no human interference. One dataset that is most similar to ours is the DIW dataset proposed by Chen et al. [20]. Unlike us, they rely on human annotators to label depth for images in the wild.

[1] Project website: http://www-personal.umich.edu/~wfchen/youtube3d.

Their data collection successfully avoids the use of depth-sensing devices, but requires a tremendous amount of manual labor. Xian et al. [36] collect depth from stereo images, but is limited by the amount of stereo images available online, while our method operates on a virtually unlimited amount of Internet videos.

2.2 Multi-View Geometry

Over the years, research on multi-view geometry has made it possible to reconstruct accurate models of landmarks, cities, and even the entire world by sifting through millions of Internet images [37, 29, 28, 26, 27, 38]. However, when applied to arbitrary Internet videos, state-of-the-art reconstruction algorithms still fail in many scenarios. Without careful parameter tuning or initialization, these methods may produce inaccurate reconstructions while maintaining a low reprojection error. This makes it difficult to filter out bad reconstructions without manual human inspection. It is thus non-trivial to directly apply off-the-shelf reconstruction methods to our task. Our proposed learning-based classifier is able to distinguish a high-quality reconstruction from a bad one. To the best of our knowledge, our method is the first to address this issue.

Reconstruction in the presence of moving objects is another challenge. Methods that address this challenge usually reconstruct objects that undergo different motions separately, and then resolve the depth/scale ambiguity of different reconstructions by enforcing various constraints [39, 40, 41]. Although they can produce reasonable depth predictions, their results are still far from perfect and their failure modes are not well understood. We thus refrain from using these methods to collect image-depth data.

Recently, Jiang et al. [42] propose to infer depth from optical flow from Youtube videos, but make no attempt to ensure the quality of obtained depth maps. Li et al. [43] recover depth from images of famous sites, and use heuristics and semantic segmentations to find good reconstruction. Our method is different in that we guarantee the quality of the reconstruction through a quality assessment network, and reconstruct depth from YouTube videos, which are more challenging than images.

3 Structure-from-Motion in the Wild

An overview of our data collection pipeline is shown in Fig. 1. For a given video, we extract and track image features, and then perform two-view reconstruction. Each reconstruction is checked by our Quality Assessment Network. The reconstructions that pass our quality check are then used to produce image-depth annotations.

3.1 Two-view Reconstruction

To obtain image-depth pairs from any given video, we can ideally perform SfM and multiview-stereo to obtain dense metric depth maps for every frame. However, as mentioned in Section 1, multiple factors can cause even the most robust SfM systems to fail on arbitrary YouTube videos.

Nevertheless, to train single-image depth estimation networks, image-depth data in the form of a dense metric depth map is not a necessity. Recently, Chen et al. [20] and Zoran et al. [44] have demonstrated that it is feasible to train depth estimators with the supervision of *relative depth* annotations, i.e., which of two pixels is closer to the viewer. By training on a sparse set of such annotations, their methods are able to produce smooth and high-quality depth estimates. Inspired by their work, we propose to construct an image-depth dataset that is accurate up to depth ordering instead of being metrically accurate. That is, we construct an image-depth dataset with relative depth annotations. This simplification alleviates the need for the optimal intrinsic camera parameters, as long as the reconstruction is accurate up to depth ordering. More concretely, we only perform SfM but skip the multview stereo, and use sparse reconstructions from SfM to annotate relative depth.

Our pipeline starts by sampling from matched features. For a given image sequence with K images, we randomly sample two arbitrary frames I_1 and I_2 along with the feature matches between them. Next we need a robust estimate of the fundamental matrix F. We proceed in a RANSAC fashion to find a set of N matches $M = \{(x_i, x'_i)\}_{i=1...N}$ that satisfies $x'^T F x = 0$, where F is estimated from the set M. This operation takes care of feature mismatches or features lying on objects undergoing different rigid motions, and leads to a more robust estimate of the fundamental matrix. Upon finding a group of N feature matches and their F, we then sample different focal lengths f and perform triangulation (we assume the camera center c_x and c_y to be the center of the image). The reconstruction R that has the smallest reprojection error is kept as the final result.

3.2 The Quality Assessment Network (QANet)

Reconstructions obtained with a simple pipeline as described above are not guaranteed to be perfectly accurate in terms of depth order. Unsuccessful reconstruction can be caused by coarse camera parameters (f, c_x, c_y), or inaccurate feature locations (subpixel localization inaccuracy during keypoint extraction), etc. Given that our goal is to build an image-depth dataset that is accurate up to depth ordering, what matters most is the percentage of point pairs that have correct ordinal depth relation. In our method, we use this percentage as a quality score $(0\% - 100\%)$ for each reconstruction. We want to automatically identify the reconstructions with a high quality score. But without the ground truth depth, it is not trivial to measure this quality score without human inspection. Besides the reconstruction itself, the only additional cues are the intermediate outputs from SfM, such as 2D feature coordinates, the fundamental matrix and the camera parameters. One possible option is to only accept reconstructions with a very small reprojection error, but this is problematic because we found that even an inaccurate reconstruction can have a small reprojection error.

Is it possible to examine the reconstruction along with its intermediate outputs to directly estimate a quality score? Here we propose the *Quality Assessment Network* (*QANet*) to achieve this goal. The intuition behind the QANet is

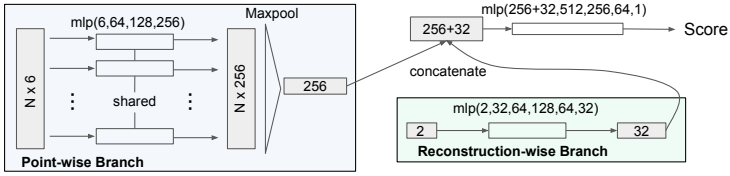

Fig. 2. Architecture of the Quality Assessment Network (QANet)

that the quality of a reconstruction is closely related to the point-wise 2D/3D locations of a reconstruction and other intermediate outputs from the SfM system. To evaluate a reconstruction, the proposed network should be able to extract useful information from point-wise features, such as the 2D projection and 3D locations of each point, as well as reconstruction-wise features, such as the overall reprojection error and other intermediate outputs. To this end, we design a network as shown in Fig 2. It consists of two branches. The *reconstruction-wise branch* is made up of multiple fully connect layers to process features such as the focal length used in the two-view reconstruction and the overall reprojection error. The *point-wise branch* processes features associated with each reconstructed point, which include (1) 2D coordinates on each of the two images, (2) the Sampson distance for the fundamental matrix recovered by the reconstructed camera rotation and translation, (3) the angle formed by the reconstructed 3D point and the two cameras. As a set of points is inherently orderless, this branch should produce consistent output regardless of their arrangement. We therefore employ an architecture similar to that of [45]. In this architecture, features for each point are independently processed by a multilayer perceptron with weights shared across points, followed by max pooling to aggregate the responses. This design keeps the output of this branch constant regardless of the ordering of the points as proven in [45]. Finally, the last stage of the network concatenates outputs from both branches and feeds the results into a series of fully connected linear layers to produce a final quality score.

The intuition behind the choice of features and the design of the two branches is that we wish to examine a reconstruction as an entity rather than a group of unrelated points. Features including the Sampson distance and the angle formed by a point and the two cameras are obtained by solving a global SfM problem and implicitly contain useful information about the entire reconstruction. Using these point-wise features allows each point to contribute to an understanding of the entire reconstruction. The reconstruction-wise branch introduces supplementary information, and as will be discussed in Section 4.1, we find the features from both branches to be instrumental in identifying high-quality reconstructions.

3.3 Training the QANet

Training the QANet requires reconstructions and their ground-truth quality scores as supervision. Here we choose to generate such training data from RGB-D video datasets, which can come from depth sensors or graphics engines.

When choosing such videos, there are a few factors to consider: (1) The video should be full of textures to generate abundant and evenly distributed feature matches between frames so that SfM can be performed; (2) The camera and object motions featured in the video should be diverse enough to mimic those in arbitrary Internet videos, so that the trained QANet can generalize well in practice.

Many RGB-D datasets like the NYU Depth [2] should satisfy condition (1). However, condition (2) may not necessarily be true. For example, the NYU Depth only captures static indoor scenes, many of which feature a camera moving forward or rotating horizontally. There is little variation in camera motion within this dataset, while for Internet videos, the camera may go through a large range of yaw, pitch and roll, as well as translation along all directions. Therefore, besides the NYU Depth dataset [2], we also use the *FlyingThings3D* [24] and the *SceneNet* [34] dataset. We choose them for the following reasons: (1) FlyingThings3D features cameras moving in random motions, as well as objects moving in random trajectories. SceneNet depicts thousands of indoor scenes where the cameras move in random trajectories while all the scene objects remain static. The camera and object motions featured in these two datasets are a close imitation of how cameras and objects move in YouTube videos; (2) Both datasets use random textures to render objects, which are rich in features. Image keypoints extracted from these textures are numerous and evenly distributed on the entire image, from which we can generate plenty of training instances and are not subjected to biases in keypoint locations. In our experiment, we leverage all three datasets and generate 350,000 training samples from each of them. The images pairs and feature matches are randomly sampled.

With the training data, we proceed to train the QANet. Ideally, the predicted quality score s' by the QANet should be exactly the same as the ground-truth score s. However, if we directly penalize the square root difference between s' and s, we notice that the network always converges to the mean of the ground-truth scores of the entire training set. To avoid this trivial solution, we change the training objective from predicting the correct scores to predicting the correct score ranking. For two reconstructions whose ground truth scores are s_1 and s_2, their predicted scores are s'_1 and s'_2, we require the ordinal relation between s'_1 and s'_2 to be the same as that of s_1 and s_2. More concretely, the loss function $h(s'_1, s'_2, s_1, s_2)$ to be optimized is:

$$h(s'_1, s'_2, s_1, s_2) = \begin{cases} \ln\left(1 + \exp(s'_2 - s'_1)\right), \text{ if } s_1 > s_2 \\ \ln\left(1 + \exp(s'_1 - s'_2)\right), \text{ if } s_1 < s_2 \end{cases} \tag{1}$$

This loss function is essentially the same as the ranking loss used by Chen et al. [20]. The intuition of this loss is to make the difference between s'_1 and s'_2 as large as possible, while keeping the sign of $(s'_1 - s'_2)(s_1 - s_2)$ positive. This loss enforces that the network produce scores that are accurate up to ranking.

We rearrange our training data according to this objective. The pool of training samples are randomly formed into pairs. We only take in sample pairs $\{(R_1, s_1), (R_2, s_2)\}$ whose absolute ground truth score difference $|s_1 - s_2|$ is greater

than a certain threshold ϕ (ϕ is set to be 5% in our experiments). We take this measure to maximize the difference between the two samples in a pair and improve training efficiency. In total, we sample one million such pairs.

3.4 Generating Relative Depth from Videos with QANet

The QANet assigns a score to every reconstruction from the two-view reconstruction pipeline. We only retain the reconstruction whose score is larger than a certain threshold ψ, whose value is determined by a validation set. The valid reconstructions are then converted into relative depth annotations. We also apply simple data augmentations to image-depth data generated, including cropping, resizing and rotating the images while changing the coordinates of the depth pairs accordingly.

When processing a video, we first employ a simple statistical approach [46] to perform shot-detection and roughly divide each video into multiple shots. We keep only one shot per video to increase the diversity of the dataset. When dealing with Internet videos, we notice some special cases which are not modeled in the data used for training QANet and may cause the QANet to fail. These cases produce unideal reconstructions which get a high score from QANet. They include when: (1) extracted features lie on subtitles and water marks; (2) the video is a slideshow (with animation); (3) the camera is static; (4) the camera zooms in or out. We deal with cases *(1)-(3)* by rejecting reconstructions whose 2D feature matches in the two images have little to no displacement, or can be modeled by a simple similarity transformation. We deal with case *(4)* by roughly estimating the ratio of focal lengths of the two images with method of [47] and rejecting those with a large ratio. We find these additional heuristics helpful in obtaining higher quality reconstructions.

4 Experiments

In this section, we first evaluate the effectiveness of the QANet in identifying good reconstructions. Next we apply the proposed data collection pipeline first on the NYU Depth dataset and then on random YouTube videos, and demonstrate that the pipeline is able to collect high-quality image-depth data to train networks that produce good depth predictions.

4.1 QANet

We first demonstrate that QANet is able to identify high-quality reconstructions. The QANet in this experiment is trained on a combination of samples from NYU Depth, FlyingThings3D and SceneNet, and then evaluated on test samples from NYU Depth and FlyingThings3D. The test samples in these two datasets are from different scenes than those used when training QANet.

In addition to training the standard QANet described in Section 3, we conduct ablative studies to identify the importance of each point-wise feature. We experiment with three versions of the QANet: (1) only using 2D coordinates

(*2D*), (2) using 2D coordinates and the angle between the 3D point and the two cameras (*2D+Ang*) (3) same as *(2)* plus Sampson distances (*2D+Ang+Sam*), i.e., the full QANet. Two additional versions of the QANet are trained to study the contribution of the two branches: (1) only has Point-wise Branch (*PointBranch*), (2) only has Reconstruction-wise Branch (*ReconBranch*). We also train a completed QANet but without the focal length in the reconstruction-wise branch (*2D+Ang+Sam-Focal*) to study the importance of reconstruction-wise features.

Recall that the QANet predicts quality scores that are accurate up to ranking. To evaluate the performance of the QANet, we measure the accuracy of the top-ranking reconstructions picked out by the QANet. As each reconstruction is associated with a ground-truth score that denotes the percentage of point pairs having correct ordinal depth, we take the average ground-truth score of the top-ranking reconstructions as their accuracy measurement. We plot this measurement of the top $n\%$ reconstructions (precision) versus top $n\%$ (recall). Such a curve depicts how accurate the top $n\%$ reconstruction picked out by the QAnet really are, and can be regarded as a pseudo-precision-recall curve. As a baseline, we also show the curve (*Groundtruth*) for the case in which the QANet is making the perfect ranking estimation for each reconstruction. We do this by ranking the test samples according to their *ground-truth* quality score, and plotting the average ground-truth score of top $n\%$ samples versus top $n\%$. We show the plots of all versions of QANet on the NYU depth and the FlyingThings3D test set in Fig. 3 and Fig. 4.

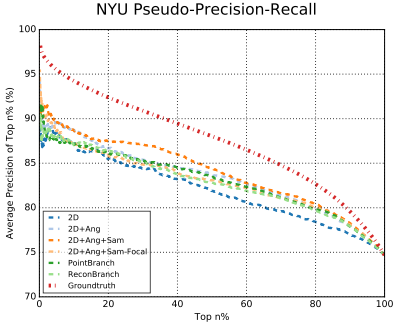

Fig. 3. The pseudo-precision-recall curve on the NYU dataset

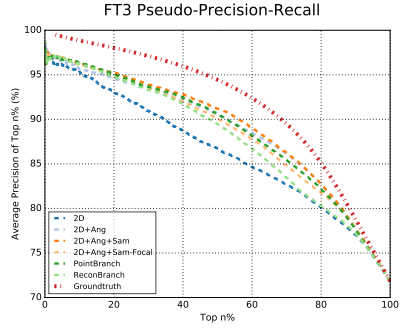

Fig. 4. The pseudo-precision-recall curve on the FlyingThings3D dataset

As shown in the two plots, adding the angle feature gives a large boost to the performance in both the NYU and the FlyingThings3D case (*2D+Ang* vs. *2D*). The Sampson distance feature helps the network perform much better in the NYU dataset(*2D+Ang+Sam* vs. *2D+Ang*), and also give minor boost to performance in the FlyingThings3D. The completed QANet *2D+Ang+Sam* that incorporates all features has the best overall performance among all versions in both datasets. *2D+Ang+Sam-Focal* underperforms *2D+Ang+Sam*, proving the

importance of focal length. It does much better in identifying the top 10% NYU test samples than *PointBranch*, indicating that the overall reprojection error in Reconstruction-wise branch is useful. These results suggest that the quality of a reconstruction can be reasonably predicted by examining a set of features, and further indicate that the proposed features all contribute in this regard. Dropping either the Point-wise branch or the Reconstruction-wise branch hurts performance, indicating that both branches are important. Although the best-performing network *2D+Ang+Sam* is not as good as the baseline *Groundtruth*, it performs reasonably well in identifying good reconstructions. For example, in the FlyingThings3D test set, the average ground-truth score of the entire test set is about 72%. By using *2D+Ang+Sam* to identify bad reconstructions, the average ground-truth score of the top 20% reconstructions reaches about 95%. In terms of the mAP (area under the curve) of the pseudo-precision-recall curve of FlyingThings3D, *2D+Ang+Sam* reaches 89.01%, while that of the *Groudtruth* is 91.28%. On the NYU Depth the mAP for *2D+Ang+Sam* and *Groudtruth* are 83.93% and 87.48% respectively.

4.2 Effectiveness of the Proposed Pipeline

Next, we demonstrate the effectiveness of our proposed data collection pipeline and compare it against the state-of-the-art SfM system Colmap [28]. Both our pipeline and Colmap use the same set of feature matches generated from Colmap under default parameter settings. To generate relative depth annotations from Colmap, we run SfM with default parameters and randomly sample point pairs from the resulting reconstructions. When comparing these two methods, the following factor need to be taken into consideration: Even given the same image sequence and feature matches, different methods can still produce different reconstructions, which may differ in the number of point contained, or the group of point reconstructed. Therefore, those reconstructions are not directly comparable, and it is not suitable to compare two methods by accuracy of those reconstructions. Recall that our ultimate goal is to collect a dataset to train high-performance depth prediction networks. We therefore propose to evaluate a data-collection method by how well the network performs when trained on data collected by that method. The idea is as follows: if a method is more effective, then the relative depth annotations it collects should be of better overall quality, and a depth-prediction network trained on them should perform better accordingly. To measure the performance of a depth-prediction network, we use the weighted Kinect disagreement rate (*WKDR*) metric as in [20]. It measures the percentage of point pairs which have incorrect predicted depth order.

We employ the method of Chen et al. [20] to train depth-prediction networks on relative depth annotations. In their method, they propose a hourglass-shape CNN to directly regress a depth map from an image. They train the network on relative depth annotations with the following loss on the predicted depth:

$$E(z_i, z_j, r) = \begin{cases} \ln\left(1 + \exp(-z_i + z_j)\right), & \text{if } r_{i,j} \in \{>\} \\ \ln\left(1 + \exp(z_i - z_j)\right), & \text{if } r_{i,j} \in \{<\} \end{cases} \tag{2}$$

Intuitively, this loss function penalizes the difference between the ground truth depth order $r_{i,j}$ and the depth order according to the predicted depth value z_i, z_j at pixel i,j. We use the same network architecture as in [20].

The NYU Depth Dataset We first experiment on the NYU Depth dataset [2]. We split the NYU Depth training set into two sets according to scene names such that both of them contain the same number of scenes. One set is denoted as NYU_{RGBD} and contains both the RGB images and the ground-truth depth, and is used for generating training data for QANet. The other set NYU_{RGB} contains only RGB images but the ground truth depth is withheld, and is used to experiment with generating relative depth annotations. We generate three sets of relative depth annotations from NYU_{RGB} — the set NYU_{QA} is generated with our pipeline, a second set NYU_{Col} is generated with Colmap [28], and a third set NYU_{QA_Sub} which is a subset of NYU_{QA} that has the same number of images as NYU_{Col}. All three sets have a maximum number of 6,000 pairs per image.

Given that QANet is trained on NYU_{RGBD}, the set NYU_{QA} is obtained with the help of this extra data that NYU_{Col} has no access to. Thus, it may not be fair to directly compare the performance of networks that are trained on NYU_{QA} (or NYU_{QA_Sub}) and NYU_{Col} respectively. To make the comparison fair, we provide another set of relative depth annotations NYU_{net}, which is generated from the point pairs and ground truth depth used in training QANet. We then compare the performance of the same network trained on these four different training sets: (1) NYU_{net}, (2) NYU_{net} + NYU_{QA}, (3) NYU_{net} + NYU_{Col}, and (4) NYU_{net} + NYU_{QA_Sub}.

We test the networks on a test set which is generated as follows: we randomly sample 1,000 point pairs along with their depth order from each image of the official NYU test set. Half of the point pairs are completely random, the other half of them are symmetric pairs. The two points in one symmetric pair lie on the same random horizontal line and share the same distance to the center of the horizontal line. As discussed in [20], this sampling strategy is to make sure that the test set is free from the bias that a point lying lower in the image or closer to the center of the image is more likely to be closer.

Table 1. WKDR results on the NYU test set

Training Set	WKDR
NYU_{net}	24.93%
NYU_{net} + NYU_{Col}	25.83%
NYU_{net} + NYU_{QA_Sub}	23.62%
NYU_{net} + NYU_{QA}	**23.01%**

We report the performance of the networks in Tab. 1. The network trained on NYU_{net} + NYU_{QA_Sub} outperforms the baseline trained on NYU_{net} alone, indicating that the additional image-depth annotations from NYU_{QA_Sub} are clearly useful in improving the performance of the depth prediction network. It also outperforms the network trained on NYU_{net}+NYU_{Col}. The network trained on NYU_{net} + NYU_{QA} performs the best. This result indicates that the relative

Fig. 5. Example relative depth annotations in YouTube3D. A relative depth pair is visualized as two points connected by an edge, where the blue point is further away than the red point. Although our pipeline occasionally produces incorrect pairs, most of them are accurate

depth supervision from our data collection pipeline are of higher quality than that from the Colmap [28].

The Depth-in-the-Wild Dataset With the ability to automatically generate an unlimited amount of relative depth annotations, we are interested in how this data can help us advance single image depth perception in the wild.

To this end, we first process about 900,000 YouTube videos, from which we construct the YouTube3D dataset along with two other baseline datasets. The three sets are constructed as follows: (1) *YouTube3D*: This set is generated from the proposed pipeline. The QANet identifies about 2 million valid reconstructions, which span 121,054 videos, totaling 795,066 images. Some examples in YouTube3D are shown in Fig. 5. (2) *Unfiltered Set (YT_{UF})*: As a baseline, we gather a second set of reconstructions that are picked randomly from the same set of reconstructions used in building YouTube3D but without applying the QANet. We use this set as an ablative study to understand the role *QANet* plays in identifying good reconstructions. This set contains roughly the same number of point pairs and images as YouTube3D. (3) *Colmap Set (YT_{Col})*: This set comes from the Colmap baseline. We run SfM on the same set of features and matches as used in constructing the previous two sets, and obtain 647,143 Colmap reconstructions, spanning 486,768 videos. We enforce the number of images in *Colmap Set* to be the same as in the previous two datasets by randomly selecting images from Colmap reconstructions. For all three sets, as the number of point pair combination in an image scales quadratically with the number of points reconstructed, we set the upper bound of point pairs per image to be 6,000 per image.

We then train models from scratch on YouTube3D, YT_{UF} and YT_{Col}. We also experiment with pretraining a network on the Full NYU Depth dataset as in [20] (*NYU*), and fine-tune it separately on YouTube3D (*NYU+YouTube3D*), the Unfiltered Set (*NYU+*YT_{UF}) and the Colmap Set (*NYU+*YT_{Col}). To test how introducing FlyingThings3D and SceneNet affects the performance, we also fine-tune *NYU* [20] on ground-truth relative depth annotations from data used for training the QANet (*NYU+FT3_SceneNet*).

A good benchmark to evaluate the performance of these networks is the Depth-in-the-Wild (DIW) dataset [20]. It consists of 74,000 test images with one manually-labeled relative depth annotation per image. However, the differences between YouTube3D and DIW are worth noting: **(1)** Images in DIW are

high-quality images from Flickr taken with a still camera. They are different from YouTube video frames in terms of the content depicted, image quality, and the way a image/video is taken, etc. **(2)** SfM cannot reconstruct objects that undergo different motions, and therefore depth order between moving objects and static backgrounds is not provided in YouTube3D. The DIW test set does not suffer from this problem because it is annotated by humans. This domain shift can be mitigated by fine-tuning *NYU+YouTube3D* on the DIW train set, which contains 420,000 training images collected by expensive manual labor (*NYU+YouTube3D+DIW*).

We show the results of all the models on the DIW test set in Tab. 2, and compare them with the state-of-the-art result *NYU+DIW* [20], which was achieved by first pre-training a model on the Full NYU Depth dataset, and then fine-tuning on the DIW training set.

Table 2. Performance of different models on the Depth-in-the-Wild (DIW) test set

Training Sets	WKDR
NYU [20]	31.31%
DIW [20]	22.14%
YT_{Col}	37.86%
YT_{UF}	25.06%
YouTube3D	**19.01%**
NYU + DIW [20]	14.39%
NYU + FT3_SceneNet	31.63%
NYU + YT_{Col}	32.25%
NYU + YT_{UF}	24.04%
NYU + YouTube3D	18.01%
NYU + YouTube3D + DIW	**13.50%**

Given the discussed domain difference, it would be difficult for networks trained purely on YouTube3D to outperform those trained on DIW directly. Nevertheless, we still see strong performance on networks trained on YouTube3D. Training from scratch on YouTube3D already significantly outperforms the baseline *NYU* [20], and beating a network trained directly on DIW train set (*DIW* [20]). By fine-tuning *NYU* [20] on YouTube3D, we gain a further performance boost (*NYU+YouTube3D*). Leveraging all data gives the best performance (*NYU+YouTube3D+DIW*), outperforming the state-of-the-art method of *NYU+DIW* [20]. These results suggest that the proposed pipeline can generate high-quality image-depth data to advance depth perception. Such results are noteworthy, especially since our dataset is gathered by a completely automatic pipeline, while the DIW training set requires extensive manual labor. These results indicate that leveraging both the human-annotated and auto-generated data is a feasible and effective way to improve single-image depth perception in the wild.

Note that *NYU+YouTube3D* outperforms *NYU+*YT_{UF}, indicating that the *QANet* is critical in identifying high-quality reconstructions. The network *NYU+FT3_SceneNet* performs on par with *NYU*, demonstrating that the performance gain of *NYU+YouTube3D* over *NYU* comes from the introduction of YouTube3D rather than FlyingThings3D and SceneNet. The network *NYU+*YT_{Col} performs

far worse than *NYU*, suggesting that the data from Colmap has a lot of noise and error and is of much lower quality than data from our pipeline.

We achieved state-of-the-art performance on the DIW test set [20] by training a network with both the DIW training set and the YouTube3D dataset. Although high-quality human annotations as in the DIW training set can help train state-of-the-art networks, they are expensive to obtain and not scalable. To further study how YouTube3D can help in the scenario that human annotations are too expensive to acquire, we fine-tune model *NYU+YouTube3D* with various amounts of the DIW training set to observe their performance. As a baseline, we also fine-tune *NYU* [20] on the same amounts of data. We report their performance in Tab. 3.

Table 3. WKDR of various models on the DIW test set. These models are fine-tuned on different percentage of the DIW training set

Percentage of DIW Training Set	Pre-trained Model	
	NYU [20]	*NYU + YouTube3D*
0%	31.31% [20]	18.01%
25%	17.89%	16.02%
50%	15.85%	14.72%
75%	15.26%	13.86%
100%	**14.39%** [20]	**13.50%**

We make the following observations: (1) The more human annotations the better the performance is. (2) Pretraining on *NYU+YouTube3D* consistently outperforms pretraining on *NYU*. As the amount of data increases, the benefit of pretraining on *NYU+YouTube3D* is less notable, and the boost provided by more training data gradually diminishes. (3) Nevertheless, fine-tuning *NYU+YouTube3D* on only 50% can already achieve similar performance to *NYU+DIW*, and fine-tuning on only 75% already outperforms *NYU+DIW*. These results indicate that when the amount of human annotation is scarce, image-depth data obtained by the proposed pipeline might be a good supplement.

5 Conclusion

In this paper we propose a fully automatic and scalable pipeline for collecting image-depth annotations from Internet videos. The pipeline uses a Quality Assessment Network to find high-quality reconstructions and produce relative depth annotations. We apply the proposed pipeline on different datasets and show that it is able to collect image-depth data which help train depth-prediction networks to achieve competitive performance in the wild.

6 Acknowledgment

This publication is based upon work partially supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-2015-CRG4-2639.

References

1. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11) (2013) 1231–1237
2. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, Springer (2012) 746–760
3. Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: Factoring shape, pose, and layout from the 2d image of a 3d scene. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
4. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
5. Roy, A., Todorovic, S.: Monocular depth estimation using neural regression forest. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5506–5514
6. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1119–1127
7. Hane, C., Ladicky, L., Pollefeys, M.: Direction matters: Depth estimation with a surface normal classifier. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 381–389
8. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5162–5170
9. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2800–2809
10. Xie, J., Girshick, R., Farhadi, A.: Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In: European Conference on Computer Vision, Springer (2016) 842–857
11. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658
12. Li, J., Klein, R., Yao, A.: Learning fine-scaled depth maps from single rgb images. arXiv preprint arXiv:1607.00730 (2016)
13. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. arXiv preprint arXiv:1702.02706 (2017)
14. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. arXiv preprint arXiv:1704.02157 (2017)
15. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 239–248
16. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 5. (2017)
17. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. (2017)

18. Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. arXiv preprint arXiv:1708.05375 (2017)
19. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: SfM-Net: Learning of structure and motion from video. arXiv preprint arXiv:1704.07804 (2017)
20. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems. (2016) 730–738
21. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. arXiv preprint arXiv:1702.04405 (2017)
22. Chang, A., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
23. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), ed.: European Conf. on Computer Vision. Part IV, LNCS 7577, Springer-Verlag (October 2012) 611–625
24. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4040–4048
25. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. arXiv preprint arXiv:1709.07322 (2017)
26. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10) (2011) 105–112
27. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3287–3295
28. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
29. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision. (2016)
30. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**(5) (2015) 1147–1163
31. Schöps, T., Schönberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proc. CVPR. Volume 3. (2017)
32. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network. In: European Conference on Computer Vision, Springer (2016) 322–337
33. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. arXiv preprint arXiv:1611.08974 (2016)
34. McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. arXiv preprint arXiv:1612.05079 (2016)
35. Zhang, Y., Song, S., Yumer, E., Savva, M., Lee, J.Y., Jin, H., Funkhouser, T.: Physically-based rendering for indoor scene understanding using convolutional neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2017) 5057–5065

36. Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., Luo, Z.: Monocular relative depth perception with web stereo data supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 311–320
37. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 1434–1441
38. Resch, B., Lensch, H.P., Wang, O., Pollefeys, M., Sorkine-Hornung, A.: Scalable structure from motion for densely sampled videos. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, IEEE (2015) 3936–3944
39. Ranftl, R., Vineet, V., Chen, Q., Koltun, V.: Dense monocular depth estimation in complex dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4058–4066
40. Russell, C., Yu, R., Agapito, L.: Video pop-up: Monocular 3d reconstruction of dynamic scenes. In: European conference on computer vision, Springer (2014) 583–598
41. Kumar, S., Dai, Y., Li, H.: Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. arXiv preprint arXiv:1708.04398 (2017)
42. Jiang, H., Learned-Miller, E., Larsson, G., Maire, M., Shakhnarovich, G.: Self-supervised depth learning for urban scene understanding. arXiv preprint arXiv:1712.04850 (2017)
43. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. arXiv preprint arXiv:1804.00607 (2018)
44. Zoran, D., Isola, P., Krishnan, D., Freeman, W.T.: Learning ordinal relationships for mid-level vision. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 388–396
45. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593 (2016)
46. Castellano, B.: Pyscenedetect. `https://github.com/Breakthrough/PySceneDetect`
47. Hartley, R.: Extraction of focal lengths from the fundamental matrix. Unpublished manuscript (1993)