# Learning Single-Image 3D from the Internet

by

Weifeng Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2020

Doctoral Committee:

Assistant Professor Jia Deng, Co-chair
Assistant Professor David Fouhey, Co-chair
Assistant Professor Justin Johnson
Professor Qiaozhu Mei

Weifeng Chen

wfchen@umich.edu

ORCID iD: 0000-0003-3352-0064

To those I met on the journey.

# ACKNOWLEDGMENTS

First and foremost, I am most grateful to my advisor, Professor. Jia Deng. I am incredibly lucky to have him as my advisor. He is always insightful and always full of great ideas that never fail to amaze me. I would like to thank him for his generous mentorship and guidance throughout my Ph.D. program. He is an inspiration to me and a role model that I look upon to.

I would also like to thank Professor. David Fouhey, Professor. Justin Johnson, and Professor. Qiaozhu Mei, for their service in the dissertation committee.

I also want to thank my family, especially my parents and my wife, for their unfailing support during the past five years. I am most grateful to my wife, Yeqian, who is always attentive, caring, and supportive, and is always there for me through all the ups and downs. Her accompany is always a source of courage and happiness for me. I am also extremely grateful for the effort and sacrifice she made in coming to Princeton with me during the last two years of my Ph.D. study. I would not have gotten this far without her support, love, and sacrifice.

Finally, I would like to thank all my friends, colleagues, collaborators, and mentors for their help and support. I learned a lot from them.

# TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE

# LIST OF TABLES

TABLE

# ABSTRACT

Single-image 3D refers to the task of recovering 3D properties such as depth and surface normals from an RGB image. It is one of the fundamental problems in Computer Vision, and its progress has the potential to bring major advancement to various other fields in vision. Although significant progress has been made in this field, the current best systems still struggle to perform well on arbitrary images "in the wild", i.e. images that depict all kinds of contents and scenes. One major obstacle is the lack of diverse training data. This dissertation makes contributions towards solving the data issue by extracting 3D supervision from the Internet, and proposing novel algorithms to learn from Internet 3D to significantly advance single-view 3D perception.

First, we have constructed "Depth in the Wild" (DIW), a depth dataset consisting of 0.5 million diverse images. Each image is manually annotated with randomly sampled points and their relative depth. After benchmarking state-of-the-art single-view 3D systems on DIW, we found that even though current arts perform well on existing datasets, they perform poorly on images in the wild. We then propose a novel algorithm that learns to estimate depth using annotations of relative depth. Compared to the state of the art, our algorithm is simpler and performs better. Experiments show that our algorithm, combined with existing RGB-D data and our new relative depth annotations, significantly improves single-image depth perception in the wild.

Second, we have constructed "Surface Normals in the Wild" (SNOW), a dataset with 60K Internet images, each manually annotated with the surface normal for one randomly sampled point. We explore advancing depth perception in the wild using surface normal as supervision. To train networks with surface normal annotations, we propose two novel losses, one that emphasizes depth accuracy, and another one that emphasizes surface normal accuracy. Experiments show that our approach significantly improves the quality of depth estimation in the wild.

Third, we have constructed "Open Annotations of Single-Image Surfaces" (OASIS), a large-scale dataset for single-image 3D in the wild. It consists of pixel-wise reconstructions of 3D surfaces for 140K randomly sampled Internet images. Six types of 3D properties are manually annotated for each image: occlusion boundary (depth discontinuity), fold boundary (normal discontinuity), surface normal, relative depth, relative normal (orthogonal, parallel, or neither), and planarity (planar or not). The rich annotations of human 3D perception in OASIS open up new research opportunities on a spectrum of single-image 3D tasks — they provide in-the-wild ground

truths either for the first time, or at a much larger scale than prior work. By benchmarking leading deep learning models on a variety of 3D tasks, we observe a large room for performance improvement, pointing to ample research opportunities for designing new learning algorithms for single-image 3D.

Finally, we have constructed "YouTube3D", a large-scale dataset with relative depth annotations for 795K images, spanning 121K videos. YouTube3D is collected fully automatically with a pipeline based on Structure-from-Motion (SfM). The key component is a novel Quality Assessment Network that identifies high-quality reconstructions obtained from SfM. It successfully eliminates erroneous reconstructions to guarantee data quality. Experiments demonstrate that YouTube3D is useful in advancing single-view depth estimation in the wild.

# CHAPTER 1

# Introduction

## 1.1 Perceiving 3D from A Single Image in the Wild

Humans have the remarkable ability to perceive 3D from visual inputs. From simple observation, we can effortlessly figure out the geometries of objects we see, including the orientations of surfaces (surface normals), the occlusion between objects (occlusion relations), the physical connectedness among surfaces and objects, and the overall geometric variation of the scene (shape). Such an ability to visually perceive 3D is indispensable to our survival in the physical world, and it would be reasonable to expect human-level artificial intelligence to possess a similar ability.

In fact, endowing machines with this ability has been a core Computer Vision problem. In particular, the ability to perform single-view 3D, i.e. perceive 3D from a single image, is especially important due to the ever-presence of monocular images and videos. Achieving this goal is significant in at least three aspects. Firstly, visually understanding the underlying 3D scene is fundamental to visual navigation and planning. Secondly, acquiring and parsing 3D object shapes is critical in making object recognition invariant to changes in viewpoint, pose, and illumination. Thirdly, even in cases where depth sensors are present, single-view 3D is still needed to handle reflective surfaces where sensors fail.

Besides its potential benefits to vision research, single-view 3D is also useful from an application point of view. It is the key to understanding scene geometries, which is an integral part of AR applications and the key to autonomous driving. It is a prerequisite for image-editing techniques such as artificial depth-of-field, stereo effects, and image re-texturing. It makes possible fast acquisition of 3D contents, which has countless applications in movies, gaming and fast prototyping.

Single-view 3D remains challenging despite significant recent progress – although state-of-the-art methods have already excelled on existing benchmarks such as NYU Depth [91] and KITTI [33], they struggle to produce accurate outputs on arbitrary images *in the wild*. Unlike indoor or cityscape images featured in standard benchmarks, in-the-wild images are taken with no constraint on cameras, scenes, contents, and illumination, being much more diverse and challeng-


Figure 1.1: Example images from current RGB-D datasets and the Depth in the Wild (DIW) dataset. Compared to images in NYU, KITTI and Make3D which depicts only indoor or cityscapes content, images in the wild are significantly more diverse.

ing. Fig. 1.1 best illustrates the difference in image diversity between prior work and images in the wild, where images from several well-established depth benchmarks are compared with those from Depth in the Wild, which is a novel depth benchmark presented by this dissertation.

Single-view 3D in the wild is hard. One major reason is that the problem is fundamentally under-constrained. While image formation is the process of projecting a rich 3D world onto a 2D plane, single-image 3D is the reverse of this process, where an image can be explained by an infinite number of valid 3D reconstructions. Nevertheless, this problem is not hopelessly unsolvable, because among all the possible reconstructions, only some would make sense. Therefore, single-image 3D is usually formulated as one of statistical inference, with much of the effort devoted to finding the most likely explanation by learning a statistical distribution. Unfortunately, the distribution varies from scenes to scenes and is not easily transferable. Unless having seen a huge amount of diverse data, which is currently lacking, it is hard to learn a distribution that is representative of the world, and models will always have difficulties generalizing in the wild.

There are two possible solutions to improve single-view 3D in the wild. The first one is to provide extensive 3D data in the wild. Intuitively, should we have all the images in the world as well as their 3D reconstructions, single-view 3D reduces to a simple problem of table look-up. Although this is unrealistic, the hope is that by going through diverse data, models can learn a distribution that is representative enough to generalize well in the wild. The second one is to develop more powerful networks and better learning strategies, so that models make more effective use of data and estimate better 3D. However, it would still be difficult if not impossible to train models that generalize well without seeing diverse data. Data is the bottleneck.

This dissertation focuses on single-view 3D in the wild, with the ultimate goal to emulate the human ability in perceiving 3D from arbitrary images. We will discuss the approaches we take to acquire extensive 3D supervision in the wild from the Internet, as well as strategies to learn from such data. We will also discuss future plans to advance this field in the final chapter.

## 1.2 Challenges

This section discusses the challenges faced by single-view 3D in two aspects: data and learning algorithms.

### 1.2.1 Data

Single-view 3D in the wild aims to reconstruct 3D from an arbitrary image. Intuitively, it is unrealistic to expect a model to accurately reconstruct an image depicting a person when all it has ever learned is to reconstruct a tree. In other words, a model needs to go through a large amount of data that is rich in *diversity* to generalize and truly grasp the essence of single-view 3D.

In fact, utilizing large datasets has been shown to be highly effective: the progress made in object recognition has largely been propelled by datasets like ImageNet [27] covering diverse object categories with high-quality labels. But unlike object recognition, single-image 3D has lacked an ImageNet equivalent that covers diverse scenes with high-quality 3D ground truths. Existing datasets are restricted to a narrow range of scenes such as indoor [91] or driving [33], mostly because they are collected with depth sensors, whose restrictions in operation has severely limited the diversity of curated data.

A lack of diverse data leads to two issues. First, learning on scene-specific 3D data has been shown to result in poor generalization [17]. This is expected as geometries vary significantly from scene to scene. Second, without diverse data, it is impossible to measure progress and gauge performance for 3D in the wild, while benchmarking has been shown to be essential toward the development of vision algorithms.

Solving single-view 3D needs diverse data. Obviously, depth sensor is not a viable solution. On the other hand, billions of images and videos are uploaded to the Internet on a monthly basis, depicting all sorts of contents and occasions. So far they have rarely been explored by the 3D vision community, largely due to one reason: 3D ground truth is not as straightforward to harvest from images as class labels or object segmentations. Designing ways to extract 3D from the Internet and utilizing them to advance single-view 3D is a promising direction but also presents new challenges.

### 1.2.2 Learning Algorithms

The goal of machine learning is to create models that generalize well in the wild. In the task of 3D reconstruction, the challenges lie in creating models that generalize well with changes in object pose, viewpoints, and intra-class variations, and be able to infer shapes of objects both with and without semantic categories.

First, pose, viewpoint and intra-class variation affect prediction results. Changes in each one of them should heavily affect the predicted 3D. For example, viewing the same chair from different

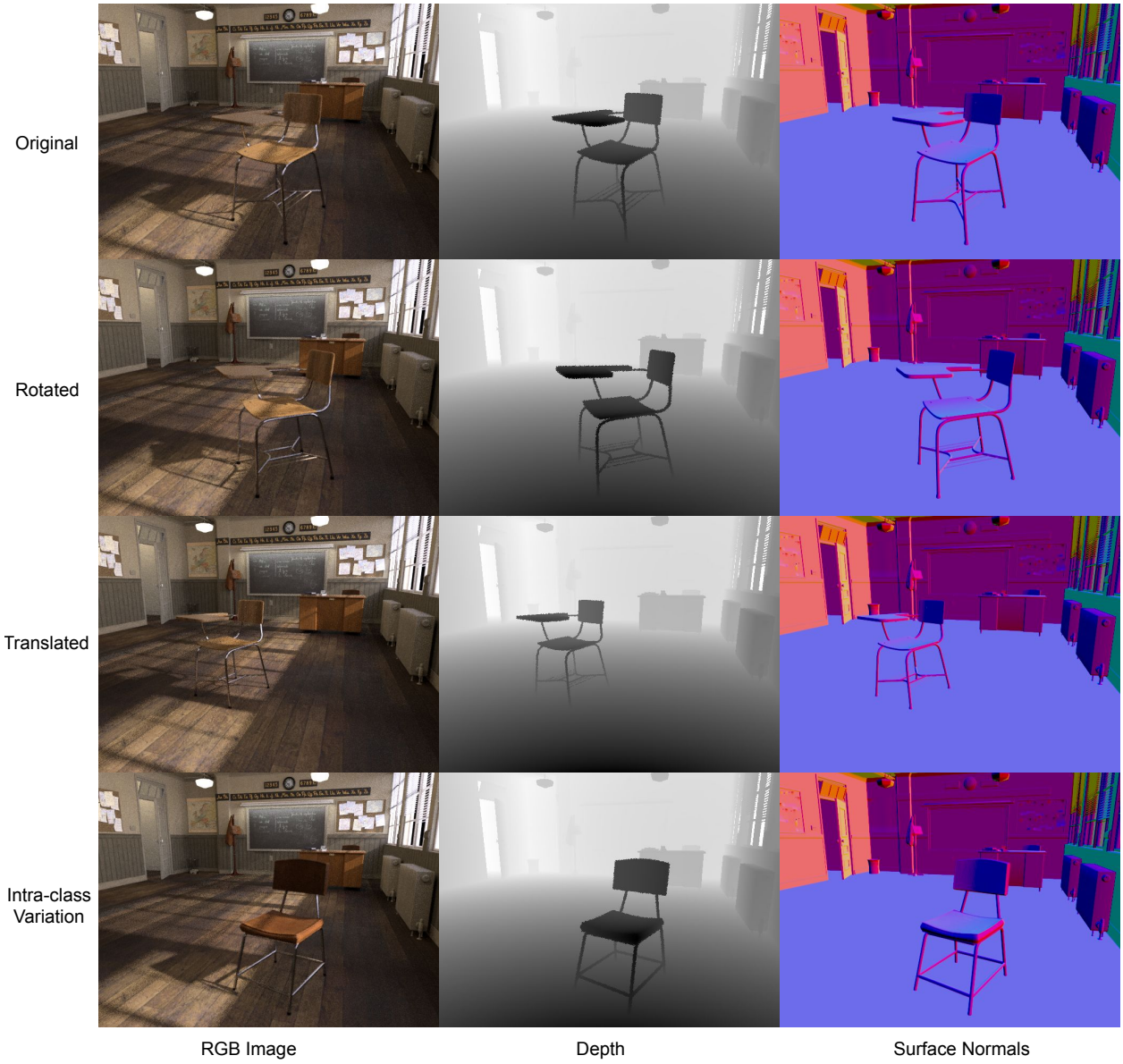



Figure 1.2: Changes in the poses, viewpoints, and positions drastically change the 3D of the chair, as evident in the depth and surface normals. Different type of chairs (intra-class variations) also leads to change in 3D.

angles results in drastically different surface normals. The same chair placed in different locations of the same scene leads to very different depth. Different designs of chairs usually have totally different shapes. Robust models must be able to predict 3D that reflects the rotation and translation an object or the scene has undergone, or even the slightest difference in designs. Fig. 1.2 best illustrates these implications.

Second, a model should not only be capable of estimating the 3D of common shapes of known object categories, but also abstract shapes of unknown object categories. The challenge of pro-

cessing abstract shapes is that no semantic information can be harnessed. In this case, a model possesses no prior knowledge of the object shapes, but instead need to rely purely on monocular 3D cues such as shading, illumination, and occlusion. Humans are able to delineate 3D structures of even the most abstract 3D sculptures, an ability that single-view 3D aims to emulate. The biological and psychological mechanism that underlies this ability is still unclear. Are current network components like convolution and skip connections enough to replicate this ability? Do networks need to process different levels of image abstractions to logically infer 3D? Or could this problem be simply solved by remembering and finding similar examples to the query image? These remain open questions that need answering in order to solve 3D reconstruction in the wild.

## 1.3 Background and Related Work

### 1.3.1 Single-view 3D

Single-view 3D has been a long-standing computer vision problem, with a large corpus of work devoted to this topic. One line of research focuses primarily on exploiting the monocular 3D cues (i.e. parallelism, orthogonality, vanishing lines and points) to automatically create 3D reconstructions [42, 60, 40, 57]. These methods are infused with human knowledge of 3D vision and make strong assumptions about the presence of 3D cues in man-made environments. They are not easily generalizable, but have relatively small or no dependence on training data.

Another line of research formulates this problem as one of statistical inference, with the primary goal of learning the mapping from 2D image space to 3D space by going through a curated set of 3D data. A dominant approach in early work is by adopting a Markov Random Field to perform statistical inference. Inference is based on multi-scale local and global image features [82, 83], and later on semantic parsing of the scene [62]. One notable exception is the work by Karsch et al. [47], which develops a non-parametric model that infers depth by finding a most similar image with known depth to the query image, and greatly improves 3D inference. Even greater advances are brought by harnessing the power of deep neural networks and large RGB-D datasets with high-quality ground truths [120, 65, 104, 28, 58, 123]. While significantly outperforming the approaches that rely on monocular 3D cues, these methods make heavy demands on the human effort to collect large datasets. Their capability to generalize depends heavily on the extent of available data.

### 1.3.2 Large-scale 3D Datasets

Recent progress in single-view 3D has been brought about by a plethora of curated datasets. Various approaches have been explored to acquire these datasets.

A dominant approach is through 3D acquisition devices such as Kinect and LIDAR. Such

devices are limited in many aspects: Kinect only functions in indoor environment, and is relatively low in precision; LIDAR, although high in precision and functions both indoor and outdoor, is notoriously expensive to operate; both devices fail on specular or transparent surfaces, and have very limited range and resolution. These limitations in sensors make it hard to collect datasets that are very diverse in scene type or content they depict. For example, NYU depth [91] consists mostly of indoor residential housing scenes with no human presence; KITTI [33] consists mostly of road scenes captured from a driving vehicle; Make3D [84] consists mostly of outdoor scenes of the Stanford campus.

Creating 3D ground truths through computer graphics is another viable option. Synthetic data enjoys the advantage of being highly accurate and can be generated easily in large quantities. It has been utilized in vision research with proven records of success. For example, the synthetic SUNCG dataset [94] has been utilized to improve single-view surface normal estimation on natural indoor images from the NYU Depth dataset [118]. The synthetic SURREAL dataset [102] has been utilized to improve human pose and shape estimation [102]. However, the diversity of synthetic data is limited by the availability of 3D "assets", i.e. shapes, materials, layouts, etc., and it remains unclear how to automatically produce data that matches the distribution of real-world data.

Human perception has also been brought in to acquire 3D. Datasets like LabelMe3D [80] and OpenSurfaces [10] crowdsource the annotation of depth and surface normals through innovative UIs. By annotating images randomly crawled from the Internet, these approaches have the potential to provide the most diverse set of 3D data. They are one of the inspirations of the work described in this dissertation. However, so far they are only able to annotate planar objects, leaving more complex geometries unexplored.

## 1.4 Contributions

This dissertation makes contributions to single-view 3D in the wild by tackling the lack of diverse training data problem. We approach it by acquiring single-view 3D ground truths from the Internet, which is a barely explored territory of 3D vision research. Contributions are made on two fronts. First, we propose scale-able methods to acquire various 3D ground truths from the Internet, and go on to present four large-scale 3D datasets, totaling 1.5 million images in the wild. Second, we propose novel ways to train single-view 3D networks on the acquired data and greatly advance 3D perception in the wild.

### 1.4.1 3D Acquisition from the Internet

What 3D information, if any, can be distilled from an image? We consider extracting depth, the most common form of 3D ground truth. And similar to the way many vision datasets are

constructed, we resort to human annotation. We observe that humans are not good at estimating metric depth [100]. Instead, they are better at perceiving qualitative aspect of depth, answering questions like "Is point A closer than point B?". Thus, we propose to annotate *relative depth*, i.e. the depth ordering between pixel pairs, through crowdsourcing on Amazon Mechanical Turk (AMT). The resulting Depth in the Wild (DIW) dataset is the first-ever relative depth dataset in the wild, consisting of 0.5 million Internet images, each annotated with randomly sampled points pairs and their relative depth (Chapter 2).

We also consider annotating surface normals. Based on the psychology study [51] that human can perceive surfaces orientation from images with striking accuracy and consistency, we crowdsource the annotation of surface normals through AMT, and construct the Surface Normals in the Wild (SNOW) dataset that consists of 60K Internet images, each having one randomly sampled point annotated with its surface normal (Chapter 3). SNOW is the first-ever dataset of crowdsourced surface normals for images in the wild.

Both DIW and SNOW are sparse annotations of 3D. We go on to design an intuitive UI that enables the pixel-wise reconstruction of depth and surface normals. This is achieved by annotating six types of 3D properties for each image: occlusion boundary (depth discontinuity), fold boundary (normal discontinuity), surface normal, relative depth (depth ordering), relative normal (orthogonal, parallel, or neither), and planarity (planar or not). We crowdsource this annotation task and introduce the Open Annotations of Single-Image Surfaces (OASIS) dataset, the first-ever large-scale dataset with dense annotations for single-image 3D in the wild, consisting of 140K images (Chapter 4).

While crowdsourcing is an effective way to obtain 3D from the Internet, automated data acquisition is a more scalable and attractive solution. We therefore propose to employ Structure-from-Motion (SfM), a technique that performs 3D reconstruction from video, to automatically harvest 3D from the Internet. We run SfM on videos randomly crawled from YouTube. To filter out erroneous SfM results and guarantee data quality, we design a Quality Assessment Network to assign a confidence score to each reconstruction, and only retain the high-quality ones to construct the YouTube3D datasets. YouTube3D is constructed in a fully automated manner, spanning 800K images from 120K random YouTube videos, with an average of 281 relative depth pairs per image (Chapter 5).

### 1.4.2 Benchmarking and Advancing Single-view 3D in the Wild

The value of the aforementioned datasets is best demonstrated by their utility in serving as benchmarks and as training resources for single-view 3D in the wild.

Compared to past benchmarks, the proposed datasets are unique as they are the first to feature images in the wild. They enable the first-ever evaluation of single-view 3D in the wild. We bench-

mark four tasks: depth estimation (on DIW and OASIS), surface normal estimation (on SNOW and OASIS), fold and occlusion boundary estimation (on OASIS), and planar surface instance segmentation (on OASIS). Our major findings are as follows: (1) models trained on scene-specific datasets often give erroneous results when presented with unfamiliar scenes with novel shapes or layouts; (2) even the state-of-the-art methods underperform human performance by a significant margin, suggesting that single-view 3D is hard; (3) standard evaluation metrics for depth and normals are limited as they often do not align well with perceptual quality. These findings are significant, as they validate the need for diverse training data in the wild, and points to new research directions. It is also worth noting that since the introduction of DIW, it has become a standard benchmark for evaluating depth estimation in the wild, and subsequently inspired several other datasets [108, 59].

The proposed datasets are also valuable training resources to advance this field, and we consider using them to improve depth and normal estimation in the wild. While training with normals in SNOW and OASIS are standard, depth ground truth in DIW, YouTube3D and OASIS are non-conventional (Chapter 2, 4, 5). We develop novel loss functions and new problem formulations to efficiently utilize such data for training. We also develop algorithms to improve depth estimation with surface normal annotations (Chapter 3). The combined effort leads to state-of-the-art methods that significantly outperform prior work in the task of single-view 3D.

# CHAPTER 2

# Single-Image Depth Perception in the Wild [1]

## 2.1 Introduction

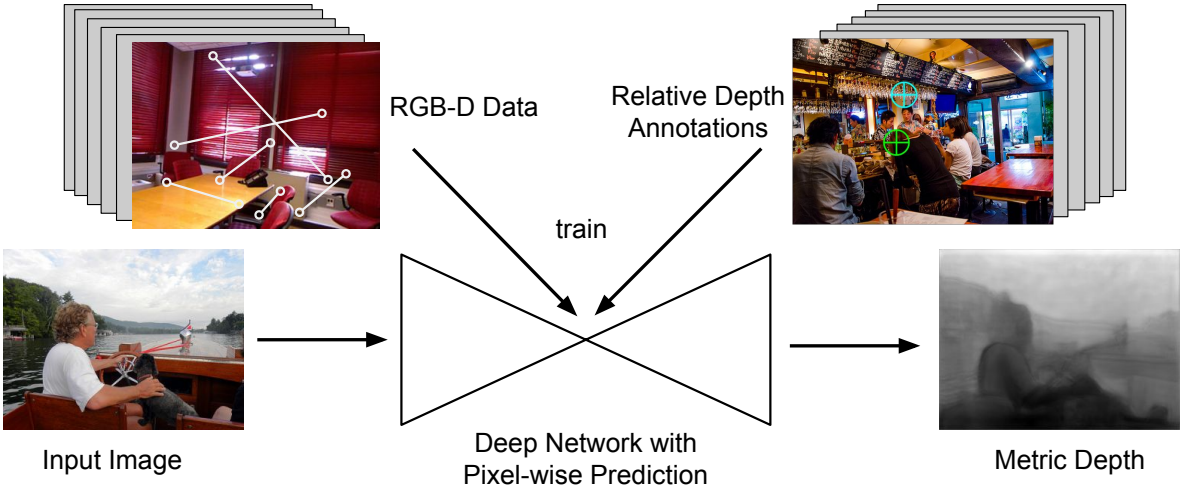



Figure 2.1: We crowdsource annotations of relative depth and train a deep network to recover depth from a single image taken in unconstrained settings ("in the wild").

Depth from a single RGB image is a fundamental problem in vision. Recent years have seen rapid progress thanks to data-driven methods [48, 40, 84], in particular, deep neural networks trained on large RGB-D datasets [91, 33, 65, 54, 28, 4, 58]. But such advances have yet to broadly impact higher-level tasks. One reason is that many higher-level tasks must operate on images "in the wild"—images taken with no constraints on cameras, locations, scenes, and objects—but the RGB-D datasets used to train and evaluate image-to-depth systems are constrained in one way or another.

Current RGB-D datasets were collected by depth sensors [91, 33], which are limited in range and resolution, and often fail on specular or transparent objects [21]. In addition, because there

[1] This chapter is based on a joint work with Zhao Fu, Dawei Yang, and Jia Deng [17].

is no Flickr for RGB-D images, researchers have to manually capture the images. As a result, current RGB-D datasets are limited in the diversity of scenes. For example, NYU depth [91] consists mostly of indoor scenes with no human presence; KITTI [33] consists mostly of road scenes captured from a car; Make3D [84, 81] consists mostly of outdoor scenes of the Stanford campus (Figure. 2.2). While these datasets are pivotal in driving research, it is unclear whether systems trained on them can generalize to images in the wild.

Is it possible to collect ground-truth depth for images in the wild? Using depth sensors in unconstrained settings is not yet feasible. Crowdsourcing seems viable, but humans are not good at estimating metric depth, or 3D metric structure in general [100]. In fact, metric depth from a single image is fundamentally ambiguous: a tree behind a house can be slightly bigger but further away, or slightly smaller but closer—the absolute depth difference between the house and the tree cannot be uniquely determined. Furthermore, even in cases where humans can estimate metric depth, it is unclear how to elicit the values from them.

But humans are better at judging relative depth [100]: "Is point A closer than point B?" is often a much easier question for humans. Recent work by Zoran et al. [123] shows that it is possible to learn to estimate metric depth using only annotations of relative depth. Although such metric depth estimates are only accurate up to monotonic transformations, they may well be sufficiently useful for high-level tasks, especially for occlusion reasoning. The seminal results by Zoran et al. point to two fronts for further progress: (1) collecting a large amount of relative depth annotations for images in the wild and (2) improving the algorithms that learn from annotations of relative depth.

In this chapter, we make contributions on both fronts. Our first contribution is a new dataset called "Depth in the Wild" (DIW). It consists of 495K diverse images, each annotated with randomly sampled points and their relative depth. We sample one pair of points per image to minimize the redundancy of annotation [2]. To the best of our knowledge this is the first large-scale dataset consisting of images in the wild with relative depth annotations. We demonstrate that this dataset can be used as an evaluation benchmark as well as a training resource [3].

Our second contribution is a new algorithm for learning to estimate metric depth using only annotations of relative depth. Our algorithm not only significantly outperforms that of Zoran et al. [123], but is also simpler. The algorithm of Zoran et al. [123] first learns a classifier to predict the ordinal relation between two points in an image. Given a new image, this classifier is repeatedly applied to predict the ordinal relations between a sparse set of point pairs (mostly between the centers of neighboring superpixels). The algorithm then reconstructs depth from the predicted ordinal relations by solving a constrained quadratic optimization that enforces additional smoothness constraints and reconciles potentially inconsistent ordinal relations. Finally, the algorithm

[2] A small percentage of images have duplicates and thus have multiple pairs.

[3] Project website: http://www-personal.umich.edu/ wfchen/depth-in-the-wild.

estimates depth for all pixels assuming a constant depth within each superpixel.

In contrast, our algorithm consists of a single deep network that directly predicts pixel-wise depth (Fig. 2.1). The network takes an entire image as input, consists of off-the-shelf components, and can be trained entirely with annotations of relative depth. The novelty of our approach lies in the combination of two ingredients: (1) a multi-scale deep network that produces pixel-wise prediction of metric depth and (2) a loss function using relative depth. Experiments show that our method produces pixel-wise depth that is more accurately ordered, outperforming not only the method by Zoran et al. [123] but also the state-of-the-art image-to-depth system by Eigen et al. [28] trained with ground-truth metric depth. Furthermore, combing our new algorithm, our new dataset, and existing RGB-D data significantly improves single-image depth estimation in the wild.

## 2.2 Related work

### 2.2.1 RGB-D Datasets

Prior work on constructing RGB-D datasets has relied on either Kinect [44, 91, 93, 22] or LIDAR [33, 84]. Existing Kinect-based datasets are limited to indoor scenes; existing LIDAR-based datasets are biased towards scenes of man-made structures [33, 84]. In contrast, our dataset covers a much wider variety of scenes; it can be easily expanded with large-scale crowdsourcing and the virually umlimited Internet images.

### 2.2.2 Intrinsic Images in the Wild

Our work draws inspiration from Intrinsic Images in the Wild [9], a seminal work that crowd-sources annotations of relative reflectance on unconstrained images. Our work differs in goals as well as in several design decisions. First, we sample random points instead of centers of superpixels, because unlike reflectance, it is unreasonable to assume a constant depth within a superpixel. Second, we sample only one pair of points per image instead of many to maximize the value of human annotations.

### 2.2.3 Depth from a Single Image

Image-to-depth is a long-standing problem with a large body of literature [8, 83, 81, 48, 65, 54, 28, 4, 58, 8, 113, 37, 62, 89, 90, 122]. The recent convergence of deep neural networks and RGB-D datasets [91, 33] has led to major advances [120, 65, 104, 28, 58, 123]. But the networks in these previous works, with the exception of [123], were trained exclusively using ground-truth metric depth, whereas our approach uses relative depth.


Figure 2.2: Example images from current RGB-D datasets and our Depth in the Wild (DIW) dataset.

Our work is inspired by that of Zoran et al. [123], which proposes to use a deep network to repeatedly classify pairs of points sampled based on superpixel segmentation, and to reconstruct per-pixel metric depth by solving an additional optimization problem. Our approach is different: it consists of a single deep network trained end-to-end that directly predicts per-pixel metric depth; there is no intermediate classification of ordinal relations and as a result no optimization needed to resolve inconsistencies.

### 2.2.4 Learning with Ordinal Relations

Several recent works [121, 71] have used the ordinal relations from the Intrinsic Images in the Wild dataset [9] to estimate surface refletance. Similar to Zoran et al. [123], Zhou et al. [121] first learn a deep network to classify the ordinal relations between pairs of points and then make them globally consistent through energy minimization.

Narihira et al. [71] learn a "lightness potential" network that takes an image patch and predicts the metric reflectance of the center pixel. But this network is applied to only a sparse set of pixels. Although in principle this lightness potential network can be applied to every pixel to produce pixel-wise reflectance, doing so would be quite expensive. Making it fully convolutional (as the authors mentioned in [71]) only solves it partially: as long as the lightness potential network has downsampling layers, which is the case in [71], the final output will be downsampled accordingly. Additional resolution augmentation (such as the "shift and stitch" approach [88]) is thus needed. In contrast, our approach completely avoids such issues and directly outputs pixel-wise estimates.

Beyond intrinsic images, ordinal relations have been used widely in computer vision and machine learning, including object recognition [73] and learning to rank [13, 45].

## 2.3 Dataset construction

We gather images from Flickr. We use random query keywords sampled from an English dictionary and exclude artificial images such as drawings and clip arts. To collect annotations of

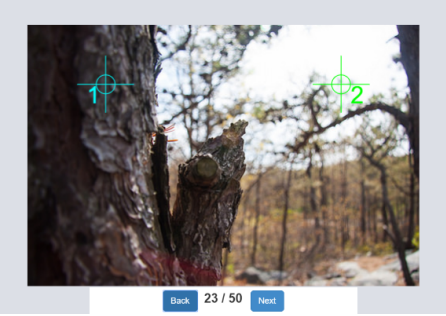



Figure 2.3: Annotation UI. The user presses '1' or '2' to pick the closer point.

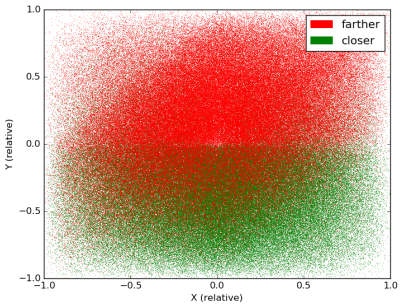



Figure 2.4: Relative image location (normalized to [-1,1]) and relative depth of two random points.

relative depth, we present a crowd worker an image and two highlighted points (Fig. 2.3), and ask "which point is closer, point 1, point 2, or hard to tell?" The worker presses a key to respond.

**How Many Pairs?** How many pairs of points should we query per image? We sample just one per image because this maximizes the amount of information from human annotators. Consider the other extreme—querying all possible pairs of points in the same image. This is wasteful because pairs of points in close proximity are likely to have the same relative depth. In other words, querying one more pair from the same image may add less information than querying one more pair from a new image. Thus querying only one pair per image is more cost-effective.

**Which Pairs?** Which two points should we query given an image? The simplest way would be to sample two random points from the 2D plane. But this results in a severe bias that can be easily exploited: if an algorithm simply classifies the lower point in the image to be closer in depth, it will agree with humans $85.8\%$ of the time (Fig. 2.4). Although this bias is natural, it makes the dataset less useful as a benchmark.

An alternative is to sample two points uniformly from a random horizontal line, which makes it impossible to use the $y$ image coordinate as a cue. But we find yet another bias: if an algorithm simply classifies the point closer to the center of the image to be closer in depth, it will agree with humans $71.4\%$ of the time. This leads to a third approach: uniformly sample two *symmetric* points with respect to the center from a random horizontal line (the middle column of Fig. 2.5). With the symmetry enforced, we are not able to find a simple yet effective rule based purely on image coordinates: the left point is almost equally likely $(50.03\%)$ to be closer than the right one.

Our final dataset consists of a roughly 50-50 combination of unconstrained pairs and symmetric pairs, which strikes a balance between the need for representing natural scene statistics and the need for performance differentiation.

**Protocol and Results:** We crowdsource the annotations using Amazon Mechanical Turk (AMT). To remove spammers, we insert into all tasks gold-standard images verified by ourselves, and reject workers whose accumulative accuracy on the gold-standard images is below 85%. We assign each


Figure 2.5: Example images and annotations. Green points are those annotated as closer in depth.

query (an image and a point pair) to two workers, and add the query to our dataset if both workers can tell the relative depth and agree with each other; otherwise the query is discarded. Under this protocol, the chance of adding a wrong answer to our dataset is less than 1% as measured on the gold-standard images.

We processed 1.24M images on AMT and obtained 0.5M valid answers (both workers can tell the relative depth and agree with each other). Among the valid answers, 261K are for unconstrained pairs and 240K are for symmetric pairs. For unconstrained pairs, It takes a median of 3.4 seconds for a worker to decide, and two workers agree on the relative depth 52% of the time; for symmetric pairs, the numbers are 3.8s and 32%. These numbers suggest that the symmetric pairs are indeed harder. Fig. 2.5 presents examples of different kinds of queries.

## 2.4 Learning with relative depth

How do we learn to predict metric depth given only annotations of relative depth? Zoran et al. [123] first learn a classifier to predict ordinal relations between centers of superpixels, and then reconcile the relations to recover depth using energy minimization, and then interpolate within each superpixel to produce per-pixel depth.

We take a simpler approach. The idea is that any image-to-depth algorithm would have to compute a function that maps an image to pixel-wise depth. Why not represent this function as a neural network and learn it from end to end? We just need two ingredients: (1) a network design

that outputs the same resolution as the input, and (2) a way to train the network with annotations of relative depth.

### 2.4.1 Network Design

Networks that output the same resolution as the input are aplenty, including the recent designs for depth estimation [28, 29] and those for semantic segmentation [66] and edge detection [111]. A common element is processing and passing information across multiple scales.

In this work, we use a variant of the recently introduced "hourglass" network (Fig. 2.6), which has been used to achieve state-of-the-art results on human pose estimation [72]. It consists of a series of convolutions (using a variant of the inception [98] module) and downsampling, followed by a series of convolutions and upsampling, interleaved with skip connections that add back features from high resolutions. The symmetric shape of the network resembles a "hourglass", hence the name. We refer the reader to [72] for comparing the design to related work. For our purpose, this particular choice is not essential, as the various designs mainly differ in how information from different scales is dispersed and aggregated, and it is possible that all of them can work equally well for our task.

### 2.4.2 Loss Function

How do we train the network using only ordinal annotations? All we need is a loss function that encourages the predicted depth map to agree with the ground-truth ordinal relations. Specifically, consider a training image $I$ and its $K$ queries $R = \{(i_k, j_k, r_k)\}, k = 1, \ldots, K$, where $i_k$ is the location of the first point in the $k$-th query, $j_k$ is the location of the second point in the $k$-th query, and $r_k \in \{+1, -1, 0\}$ is the ground-truth depth relation between $i_k$ and $j_k$: closer $(+1)$, further $(-1)$, and equal $(0)$. Let $z$ be the predicted depth map and $z_{i_k}, z_{j_k}$ be the depths at point $i_k$ and $j_k$. We define a loss function

$$L(I, R, z) = \sum_{k=1}^{K} \psi_k(I, i_k, j_k, r, z), \tag{2.1}$$

where $\psi_k(I, i_k, j_k, z)$ is the loss for the $k$-th query

$$\psi_k(I, i_k, j_k, z) = \begin{cases} \log\left(1 + \exp(-z_{i_k} + z_{j_k})\right), & r_k = +1 \\ \log\left(1 + \exp(z_{i_k} - z_{j_k})\right), & r_k = -1 \\ (z_{i_k} - z_{j_k})^2, & r_k = 0. \end{cases} \tag{2.2}$$

This is essentially a ranking loss: it encourages a small difference between depths if the ground-truth relation is equality; otherwise it encourages a large difference.

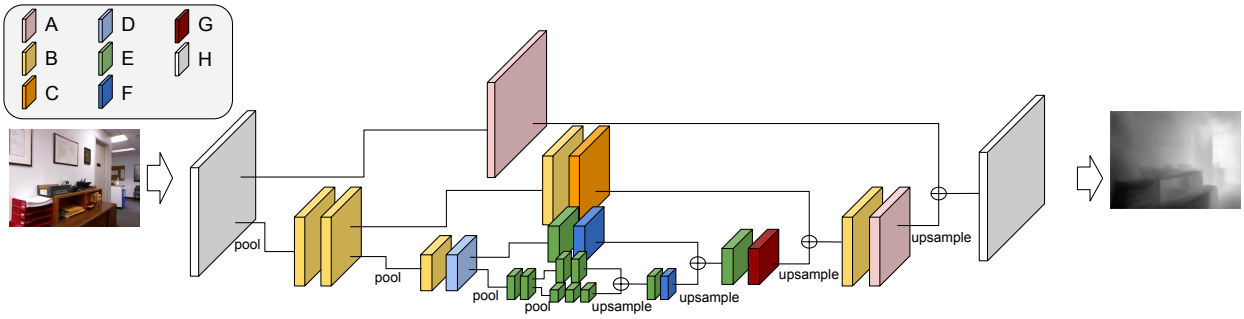



Figure 2.6: Network design. Each block represents a layer. Blocks sharing the same color are identical. The $\oplus$ sign denotes the element-wise addition. Block H is a convolution with 3x3 filter. All other blocks denote the Inception module shown in Figure 2.7. Their parameters are detailed in Tab. 2.1

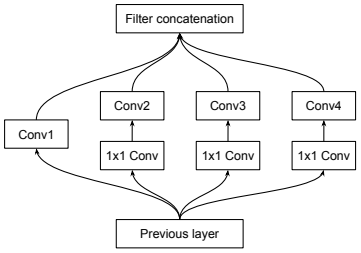



Figure 2.7: Variant of Inception Module [98] used by us.

### 2.4.3 Novelty of Our Approach

Our novelty lies in the combination of a deep network that does pixel-wise prediction and a ranking loss placed on the pixel-wise prediction. A deep network that does pixel-wise prediction is not new, nor is a ranking loss. But to the best of our knowledge, such a combination has not been proposed before, and in particular not for estimating depth.

| Block Id | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| #In/#Out | 128/64 | 128/128 | 128/128 | 128/256 | 256/256 | 256/256 | 256/128 |
| Inter Dim | 64 | 32 | 64 | 32 | 32 | 64 | 32 |
| Conv1 | 1x1 | 1x1 | 1x1 | 1x1 | 1x1 | 1x1 | 1x1 |
| Conv2 | 3x3 | 3x3 | 3x3 | 3x3 | 3x3 | 3x3 | 3x3 |
| Conv3 | 7x7 | 5x5 | 7x7 | 5x5 | 5x5 | 7x7 | 5x5 |
| Conv4 | 11x11 | 7x7 | 11x11 | 7x7 | 7x7 | 11x11 | 7x7 |

Table 2.1: Parameters for each type of layer in our network. *Conv1* to *Conv4* are sizes of the filters used in the components of Inception module shown in Figure.2.7. Conv2 to 4 share the same number of input and is specified in *Inter Dim*.


Figure 2.8: Qualitative results on NYU Depth by our method, the method of Eigen et al. [28], and the method of Zoran et al. [123]. All depth maps except ours are directly from [123].

## 2.5 Experiments on NYU Depth

We evaluate our method using NYU Depth [91], which consists of indoor scenes with ground-truth Kinect depth. We use the same setup as that of Zoran et al. [123]: point pairs are sampled from the training images (the subset of NYU Depth consisting of 795 images with semantic labels) using superpixel segmentation and their ground-truth ordinal relations are generated by comparing the ground-truth Kinect depth; the same procedure is applied to the test set to generate the point pairs for evaluation (around 3K pairs per image). We use the same training and test data as Zoran et al. [123].

| Method | WKDR | $\text{WKDR}^{=}$ | $\text{WKDR}^{\neq}$ |
|---|---|---|---|
| Ours | **35.6%** | **36.1%** | **36.5%** |
| Zoran [123] | 43.5% | 44.2% | 41.4% |
| rand_12K | **34.9%** | **32.4%** | **37.6%** |
| rand_6K | 36.1% | 32.2% | 39.9% |
| rand_3K | 35.8% | 28.7% | 41.3% |
| Ours_Full | **28.3%** | **30.6%** | **28.6%** |
| Eigen(A) [28] | 37.5% | 46.9% | 32.7% |
| Eigen(V) [28] | 34.0% | 43.3% | 29.6% |

Table 2.2: ordinal error measures (disagreement rate with ground-truth depth ordering) on NYU Depth.

| Method | RMSE | RMSE (log) | RMSE [4] (s.inv) | absrel | sqrrel |
|---|---|---|---|---|---|
| Ours | 1.13 | 0.39 | 0.26 | 0.36 | 0.46 |
| Ours_Full | 1.10 | 0.38 | 0.24 | 0.34 | 0.42 |
| Zoran [123] | 1.20 | 0.42 | - | 0.40 | 0.54 |
| Eigen(A) [28] | 0.75 | 0.26 | 0.20 | 0.21 | 0.19 |
| Eigen(V) [28] | **0.64** | **0.21** | **0.17** | **0.16** | **0.12** |
| Wang [104] | 0.75 | - | - | 0.22 | - |
| Liu [65] | 0.82 | - | - | 0.23 | - |
| Li [58] | 0.82 | - | - | 0.23 | - |
| Karsch [48] | 1.20 | - | - | 0.35 | - |
| Baig [3] | 1.0 | - | - | 0.3 | - |

Table 2.3: metric error measures on NYU Depth. Details for each metric can be found in [28]. There are two versions of results by Eigen et al. [28], one using AlexNet (Eigen(A)) and one using VGGNet (Eigen(V)). Lower is better for all error measures.

As the system by Zoran et al. [123], our network predicts one of the three ordinal relations on the test pairs: equal ($=$), closer ($<$), or farther ($>$). We report WKDR, the weighted disagreement rate between the predicted ordinal relations and ground-truth ordinal relations [5]. We also report $\text{WKDR}^{=}$ (disagreement rate on pairs whose ground-truth relations are $=$) and $\text{WKDR}^{\neq}$ (disagreement rate on pairs whose ground-truth relations are $<$ or $>$).

Since two ground-truth depths are almost never exactly the same, there needs to be a relaxed definition of equality. Zoran et al. [123] define two points to have equal depths if the ratio between their ground-truth depths is within a pre-determined range. Our network predicts an equality relation if the depth difference is smaller than a threshold $\tau$. The choice of this threshold will result in different values for the error metrics (WKDR, $\text{WKDR}^{=}$, $\text{WKDR}^{\neq}$): if $\tau$ is too small, most pairs will be predicted to be unequal and the error metric on equality relations ($\text{WKDR}^{=}$) will be large; if $\tau$ is too big, most pairs will be predicted to be equal and the error metric on inequality relations ($\text{WKDR}^{\neq}$) will be large. We choose the threshold $\tau$ that minimizes the maximum of the three error metrics on a validation set held out from the training set. Tab. 2.2 compares our network (*ours*) versus that of Zoran et al. [123]. Our network is trained with the same data [6] but outperforms [123] on all three metrics.

Following [123], we also compare with the state-of-art image-to-depth system by Eigen et al. [28], which is trained on pixel-wise ground-truth metric depth from the full NYU Depth training set (220K images). To compare fairly, we give our network access to the full NYU Depth training set. In addition, we remove the limit of 800 point pairs per training image placed by Zoran et al and use all available pairs. The results in Tab. 2.2 show that our network (*ours_full*) achieves superior

[5] WKDR stands for "Weighted Kinect Disagreement Rate"; the weight is set to 1 as in [123]

[6] The code released by Zoran et al. [123] indicates that they train with a random subset of 800 pairs per image instead of all the pairs. We follow the same procedure and only use a random subset of 800 pairs per image.

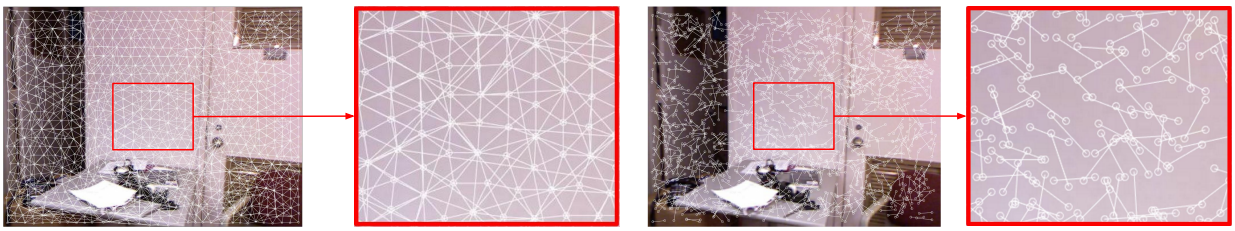


Figure 2.9: Point pairs generated through superpixel segmentation [123] (left) versus point pairs generated through random sampling with distance constraints (right).

performance in estimating depth ordering. Granted, this comparison is not entirely fair because [28] is not optimized for predicting ordinal relations. But this comparison is still significant in that it shows that we can train on only relative depth and rival the state-of-the-art system in estimating depth up to monotonic transformations.

In Figure. 2.8 we show qualitative results on the same example images used by Zoran et al. [123]. We see that although imperfect, the recovered metric depth by our method is overall reasonable and qualitatively similar to that by the state-of-art system [28] trained on ground-truth metric depth.

**Metric Error Measures.** Our network is trained with relative depth, so it is unsurprising that it does well in estimating depth up to ordering. But how good is the estimated depth in terms of metric error? We thus evaluate conventional error measures such as RMSE (the root mean squared error), which compares the absolute depth values to the ground truths. Because our network is trained only on relative depth and does not know the range of the ground-truth depth values, to make these error measures meaningful we normalize the depth predicted by our network such that the mean and standard deviation are the same as those of the mean depth map of the training set. Tab. 2.3 reports the results. We see that under these metric error measures our network still outperforms the method of Zoran et al. [123]. In addition, while our metric error is worse than the current state-of-the-art, it is comparable to some of the earlier methods (e.g. [48]) that have access to ground-truth metric depth.

**Superpixel Sampling versus Random Sampling.** To compare with the method by Zoran et al. [123], we train our network using the same point pairs, which are pairs of centers of superpixels (Fig. 2.9). But is superpixel segmentation necessary? That is, can we simply train with randomly sampled points?

To answer this question, we train our network with randomly sampled points. We constrain the distance between the two points to be between 13 and 19 pixels (out of a 320×240 image) such that the distance is similar to that between the centers of neighboring superpixels. The results are included in Tab. 2.2. We see that using 3.3k pairs per image (***rand_3K***) already achieves comparable performance to the method by Zoran et al. [123]. Using twice or four times as many

| Method | Eigen(V) [28] | Ours_Full | Ours_NYU_DIW | Ours_DIW | Query_Location_Only |
|---|---|---|---|---|---|
| WHDR | 25.70% | 31.31% | **14.39%** | 22.14% | 31.37% |

Table 2.4: Weighted Human Disagreement Rate (WHDR) of various methods on our DIW dataset, including Eigen(V), the method of Eigen et al. [28] (VGGNet [92] version)

pairs (***rand_6K***, ***rand_12K***) further improves performance and significantly outperforms [123].

It is worth noting that in all these experiments the test pairs are still from superpixels, so training on random pairs incurs a mismatch between training and testing distributions. Yet we can still achieve comparable performance despite this mismatch. This shows that our method can indeed operate without superpixel segmentation.

## 2.6 Experiments on Depth in the Wild

In this section we experiment on our new Depth in the Wild (DIW) dataset. We split the dataset into 421K training images and 74K test images [7].

We report the WHDR (Weighted Human Disagreement Rate) [8] of 5 methods in Tab. 2.4: (1) the state-of-the-art system by Eigen et al. [28] trained on full NYU Depth; (2) our network trained on full NYU Depth (Ours_Full); (3) our network pre-trained on full NYU Depth and fine-tuned on DIW (Ours_NYU_DIW); (4) our network trained from scratch on DIW (Ours_DIW); (5) a baseline method that uses only the location of the query points: classify the lower point to be closer or guess randomly if the two points are at the same height (Query_Location_Only).

We see that the best result is achieved by pre-training on NYU Depth and fine-tuning on DIW. Training only on NYU Depth (Ours_NYU and Eigen) does not work as well, which is expected because NYU Depth only has indoor scenes. Training from scratch on DIW achieves slightly better performance than those trained on only NYU Depth despite using much less supervision. Pre-training on NYU Depth and fine-tuning on DIW leaverages all available data and achieves the best performance. As shown in Fig. 2.10, the quality of predicted depth is notably better with fine-tuning on DIW, especially for outdoor scenes. These results suggest that it is promising to combine existing RGB-D data and crowdsourced annotations to advance the state-of-the art in single-image depth estimation.

[7] 4.38% of images are duplicates downloaded using different query keywords and have more than one pairs of points. We have removed test images that have duplicates in the training set.

[8] All weights are 1. A pair of points can only have two possible ordinal relations (farther or closer) for DIW.


Figure 2.10: Qualitative results on our Depth in the Wild (DIW) dataset by our method and the method of Eigen et al. [28].

## 2.7 Summary

We have studied single-image depth perception in the wild, recovering depth from a single image taken in unconstrained settings. We have introduced a new dataset consisting of images in the wild annotated with relative depth and proposed a new algorithm that learns to estimate metric depth supervised by relative depth. We have shown that our algorithm outperforms prior art and our algorithm, combined with existing RGB-D data and our new relative depth annotations, significantly improves single-image depth perception in the wild.

# CHAPTER 3

# Surface Normals in the Wild [1]

## 3.1 Introduction

In the previous chapter, we discussed estimating depth for images "in the wild": we collected human annotations of relative depth—the depth ordering of two points—for random Internet images and use the annotations to train a deep network that directly predicts metric depth. We showed that it is possible to improve depth estimation for images in the wild by using human annotations of depth. In particular, we showed that while it is difficult to obtain absolute metric depth (per-pixel depth values) from humans, it is nonetheless feasible to collect *indirect, qualitative* depth annotations such as relative depth, and use such annotations to learn to estimate metric depth. This strategy does not rely on depth sensors and can work with arbitrary images; it thus has the potential to significantly advance depth estimation in the wild.

One limitation of the work discussed in the previous chapter, however, is that annotations of relative depth do not capture all information that is perceptually important. In particular, relative depth is invariant to monotonic transformations of metric depth, meaning that there can be two scenes that are perceptually very different yet are indistinguishable in terms of relative depth. For example, it is possible to bend, wiggle, or tilt a straight line without affecting relative depth (Fig. 3.2). In other words, relative depth does not capture important perceptual properties such as continuity, surface orientation, and curvature. As a result, systems trained on relative depth will not necessarily recover depth that is perceptually faithful in all aspects.

In this chapter, we build on the previous chapter and address the limitation by introducing an additional type of indirect, qualitative depth annotation—surface normals. Surface carries important information on 3D geometry: they encode the local orientation of surfaces and the derivatives of depth. In fact, given dense surface normals, it is possible to recover full metric depth up to scaling and translation. This suggests that annotations of surface normals can eliminate the ambiguities in relative depth and result in better depth estimation. In addition, it has been well documented

[1] This chapter is based on a joint work with Donglai Xiang, and Jia Deng [20].

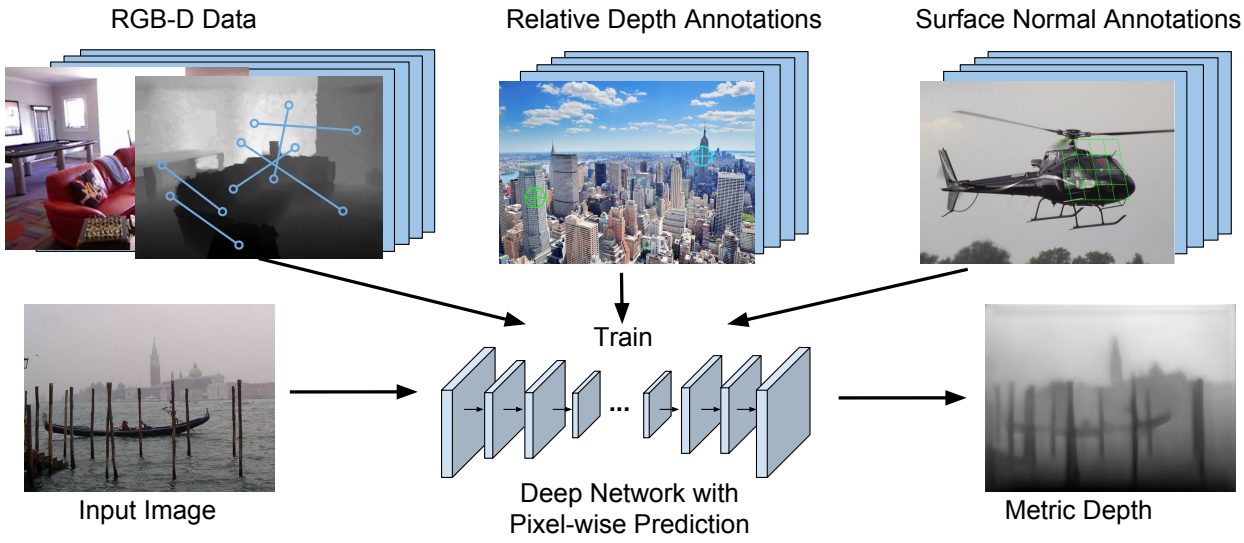



Figure 3.1: Building on top of the work of Chen et al. [17], we crowdsource annotations of surface normals and use the collected surface normals to help train a better depth prediction network.

in human vision research that humans perceive surface orientation with a remarkable degree of consistency [51]. This suggests that it could be feasible to collect human annotations for images in the wild.

We consider two questions: how to crowdsource annotations of surface normals, and how to use surface normal annotations to help train a network that predicts per-pixel metric depth. To crowdsource surface normals, we develop a UI that allows a user to annotate a surface normal by adjusting a virtual arrow and a virtual tangent plane. This UI allows human annotators to reliably estimate surface normals. With this UI we introduce a dataset called "Surface Normals in the Wild" (SNOW), which consists of surface normal annotations collected from 60,061 Flickr images.

To incorporate surface normal annotations into training, we develop two novel loss functions to train a deep network that directly predicts metric depth. The first loss function is based on directly comparing normals, that is, computing the angular difference between the ground truth normals and the normals derived from the predicted depth. The second loss function is based on comparing depth derivatives, i.e., computing the discrepancy between the derivative of the predicted depth and the derivative given by the ground truth normals. We show that each approach incurs its own trade-offs and emphases on different aspects of depth quality, and should be chosen based on particular applications.

Our main contributions are (1) a new dataset of crowdsourced surface normals for images in the wild and (2) two distinct approaches of for using surface normal annotations to train a deep network that directly predicts per-pixel metric depth. Experiments on both NYU Depth [91] and

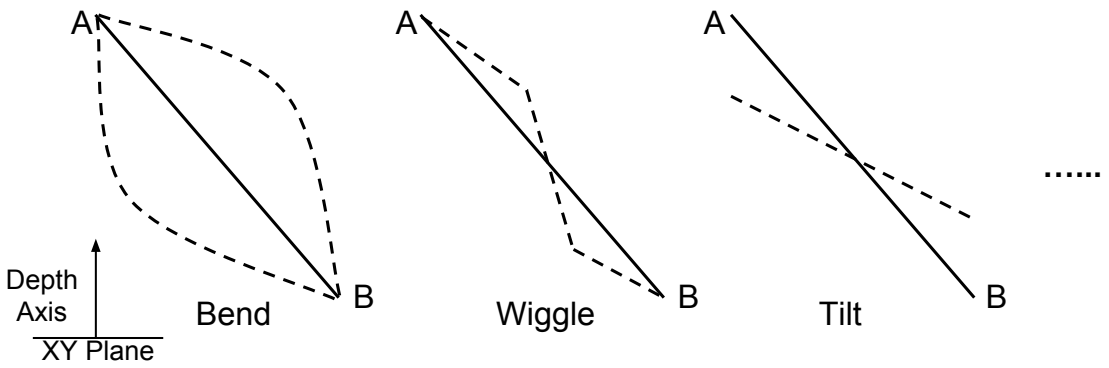



Figure 3.2: Ambiguities of relative depth annotation. Bending, wiggling, or tilting a 3D surface from solid line configuration to dotted line configuration does not change the ordinal relation that point A is farther away from the camera than point B.

SNOW demonstrate that surface normal annotations can significantly improve the quality of depth estimation.

## 3.2 Related work

### 3.2.1 Datasets with depth and surface normals

Prior works on estimating depth or surface normals have mostly used NYU Depth [91] , Make3D [83], KITTI [33], or ScanNet [26]. Although these datasets provide highly accurate depth, as pointed out by Chen et al. [17] they are limited to specific types of scenes. The same limitation applies to synthetic datasets such as MPI Sintel [12] and the dataset by [78] because the 3D content had to be manually created. The Depth in the Wild (DIW) dataset introduced by Chen et al. [17] takes a major step toward including arbitrary scenes in the wild. However, DIW provides only relative depth annotations, which lack information on many essential 3D properties such as surface normals. We build upon DIW and introduce a new dataset of crowdsourced surface normals for images in the wild.

Open Surfaces [10] is a large dataset of images with annotations of surface properties including surface normals and material. However, open Surfaces is not suitable for depth estimation in the wild: it contains only images of indoor scenes. In addition, it only has surface normals for planar surfaces, whereas our dataset has no such restriction.

### 3.2.2 Depth and surface normals from a single image

There has been a large body of work on estimating depth and/or surface normals from a single image [65, 29, 58, 4, 28, 55, 54, 104, 106, 48, 8]. All these methods use dense ground truth depth or

normals during training, except the work of Zoran et al [123] which uses relative depth for training. They all have difficulty generalizing to images in the wild due to the limited scene diversity of the existing datasets that were acquired by depth sensors.

Chen et al. [17] instead use crowdsourced relative depth for training, using indirect depth human annotations to get around the limitations of depth sensors. Our work goes beyond the work of Chen et al. by exploring surface normals.

Two other recent works [32, 110] have also leveraged indirect supervision of depth. In particular, they have used pairs of stereo images to impose constraints on the predicted depth, e.g. the depth estimated from the left image should be consistent with the depth estimated from the right image as dictated by epipolar geometry [32].

Chakrabarti et al. [14] trained a network that simultaneously predicts distributions of depth and distributions of depth derivatives at each pixel location. Then they used a global optimization method to recover a single depth map that is most consistent with the predictions. Our work differs in two ways. First, the only output of our network is a depth map. Our network does not directly predict surface normals or depth derivatives, and thus there is no need for additional optimization steps to harmonizing the outputs. Second, we do not use dense ground truth metric depth in training. Our ground truth annotations are sparse and involve only relative depth and/or surface normals.

### 3.2.3 Surface normals in 3D reconstruction

Surface normals have played important roles in many 3D reconstruction systems. For example, surface normals have been used to infer 3D models [53], create watertight 3D surfaces [49], regularize planar object reconstruction [105], and to aid multi-view reconstruction [31] and structure from motion [43], or depth estimation [37]. In our approach, surface normals are used in training only; the network directly predicts depth, without explicitly producing surface normals.

## 3.3 Dataset construction

Similar to the Depth in the Wild (DIW) dataset by Chen et al. [17], we source our images from Flickr using random keywords from an English dictionary. For each image, we extract the focal length of the camera from the EXIF metadata—the focal length is needed for determining the amount of perspective distortion when we visualize a surface normal on top of an image in our UI.

To collect surface normal annotations, we present a crowd worker with an image and a highlighted location (Fig. 3.3). The worker then draws a surface normal using a set of controls: she can pick a point on a sphere, or use two slider bars to adjust the angles (there are two degrees of

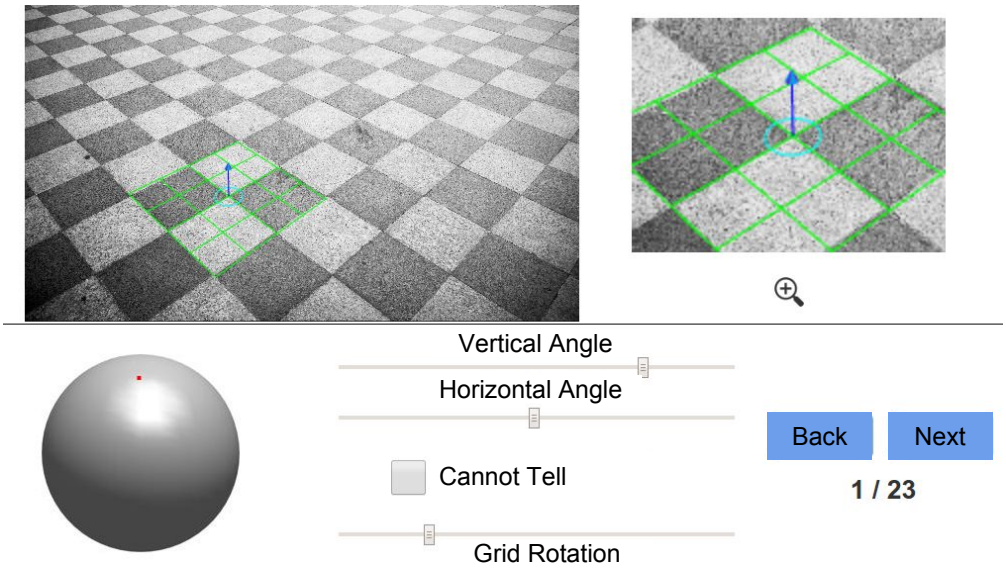



Figure 3.3: The annotation UI we use for data collection. The query image is displayed on the top left with the keypoint highlighted. A zoom-in view centered at the keypoint is displayed on the top right to help the worker see the details better. Workers then click on the sphere and adjust the slider bars to annotate the surface normal.

Figure 3.4: Some examples of the final surface normal annotations we gather for the SNOW dataset. The green grid denotes the tangent plane, and the red arrow denotes the surface normal. For best visual effect, please view in color.

freedom). The surface normal is visualized as an arrow originating from a 2D grid that represents the tangent plane. Both the arrow and the 2D grid are rendered taking into account the focal length extracted from the image metadata. This visualization is inspired by the gauge figures used in human vision research [51]; it helps the worker perceive the surface normal in 3D.

For each image, we pick one random location uniformly from the 2D plane to have its surface normal annotated. Following Chen et al. [17] we only pick one random location to minimize the correlation between annotations.

As the locations are randomly picked, some may fall onto areas where the surface normal is hard to infer, especially when there is a large amount of clutter or texture, e.g. tree leaves in the distance or grass in a field. Surface normals may also be impossible to infer on regions such as the


Figure 3.5: Some examples of the very difficult cases where the surface normal is hard to infer from the image. Point A is on tree leaves, which are small and cluttered. Point B is on a dark background where nothing can be seen clearly. In these case, the worker can indicate that the surface normal is hard to tell. Please view in color.

sky or a dark background (Fig. 3.5). In these cases a user can indicate that the surface normal is hard to tell.

We crowdsource the task through Amazon Mechanical Turk. We randomly inject gold standard samples into the task to identify spammers. Each surface normal is annotated by two different workers. If the two annotations are within 30 degree of each other, then we take the average of the two (renormalized to a unit vector) as the final annotation; otherwise, we discard both annotations.

Fig. 3.4 shows some examples of the collected normals. In total, we processed 210,000 images on Amazon Mechanical Turk and obtain 60,061 valid samples. On average, it takes about 15 seconds for a worker to annotate one surface normal. The average angular difference between the two accepted annotation is 14.32°. This suggests that human annotations usually agree with each other quite well.

### 3.3.1 Quality of human annotated surface normals

An important question is how consistent and accurate the human annotations are. To study this, we collect human annotations of surface normals on a random sample of 113 NYU Depth [91] images. Each surface normal is estimated by three human annotators. We compare the human annotations with the ground truth surface normals (derived from the Kinect ground truth depth). We measure the Human-Human Disagreement (HHD) using the average angular difference between a human annotation and the mean of multiple human annotations. We measure Human-Kinect Disagreement (HKD) using the average angular difference between a human annotation and the Kinect ground truth.

We found that the Human-Human Disagreement on our sample is ($7.4^\circ$). This suggests that human annotations are remarkably consistent between each other. However, the Human-Kinect Disagreement is $32.8^\circ$ which at first glance seems to suggest that human annotations contain a large amount of systemic bias measured against the Kinect ground truth. However, a close inspection reveals that most of the disagreement is a result of imperfect Kinect ground truth rather than biased human estimation.

One source of Kinect error is holes in the raw depth map. Some holes are due to specular or reflective surfaces; others are due to the parallax caused by the RGB camera located slightly away from the depth camera. The holes in the raw Kinect depth map are filled through some heuristic post-processing. Such hole-filling is imperfect. It is especially problematic at cluttered regions because it cannot recover the fine variations of depth and as a result the derived normals will be inaccurate.

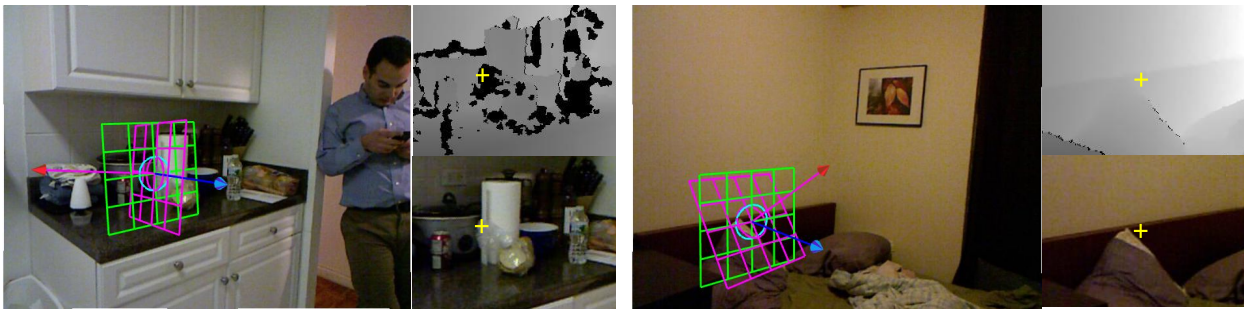
Another source of Kinect error is imperfect normals computed from accurate depth. In this experiment we used the official toolkit from the NYU Depth dataset [91] to compute normals. Each normal is computed by fitting a plane to a neighborhood of pixels. But this procedure tends to smooth out normals at or close to sharp normal discontinuities (e.g. at the intersection of two planes or at occlusion boundaries). This problem is especially severe in cluttered regions where there are many such discontinuities. But human estimation of normals is not susceptible to this issue.

We manually inspected every image in our sample and found that 37% of the cases can be attributed to one of the two sources of Kinect error (holes or imperfect normal calculation). Fig. 3.6 shows examples of such cases. The Human-Kinect disagreement on these problematic cases is $44.32^\circ$. Excluding these cases, the Human-Kinect disagreement is only $15.64^\circ$. It is worth noting that in those cases of Human-Kinect disagreement, humans remain remarkably consistent among themselves (average disagreement is $7.17^\circ$). These results suggest that human annotations of surface normals are of high quality.

It is worth noting that due to the inherent ambiguity of single-image depth estimation, we can never expect humans to match the accuracy of depth sensors, which use more than a single image to recover depth. And in many applications, especially those involving recognition, metric fidelity is not essential. Consistency is the more important quality measure because it means that there is a consistent representation (possibly biased) that we can hope to learn to estimate.

## 3.4 Learning with surface normals

Our goal is to train a deep neural network to perform depth prediction. We build our method upon [17], which uses relative depth as supervision during training. The main idea from [17]



(a) (b)

Figure 3.6: Examples of Kinect error. It shows annotations along with zoom-in views of depth map and RGB image around the keypoint (yellow cross). The red arrow with a purple mesh shows the Kinect ground-truth. Blue arrow and green mesh shows human annotations. (a) lies on a hole in the depth map which is caused by the transparent plastic bag. (b) lies near depth discontinuities. The surface normal in these region cannot be reliably computed.

is to train a network using a loss function that penalizes the inconsistency between the predicted depth and the ground truth relative depth (ordinal relations between pairs of points). We propose to incorporate surface normals as additional supervision. This translates to a loss function that encourages the predicted depth to be consistent with both the ground truth relative depth and the ground truth surface normals.

Formally, let $I$ be a training image with $K$ relative depth annotations and $L$ surface normal annotations. Using the same notations of [17], let $R = (i_k, j_k, r_k), k = 1 \dots K$ be the set of relative depth annotations, where $i_k$ and $j_k$ are the locations of two points in the $k$-th annotation and $r_k \in \{>, <, =\}$ is the ground-truth ordinal relation (closer, further, or same distance). Let $S = \{p_l, n_l\}$ be the set of surface normal annotations, where $p_l$ is the location of the $l$-th annotation and $n_l \in \mathbf{R}^3$ is the ground truth surface normal at this location.

We can now express the loss function as follows:

$$L(R, S, z) = \frac{1}{K} \sum_{k=1}^{K} \psi(i_k, j_k, r_k, z) + \lambda \frac{1}{L} \sum_{l=1}^{L} \phi(p_l, n_l, z) \tag{3.1}$$

where $z$ is the depth map predicted by the network. The loss term $\psi(i_k, j_k, r_k, z)$ measures the inconsistency between the predicted depth map $z$ and the $k$-th relative depth annotation. The loss term $\sum_{l=1}^{L} \phi(p_l, n_l, z)$ measures the inconsistency between the predicted depth map $z$ and the $l$-th surface normal annotation. The hyper-parameter $\lambda$ balances the two terms.

### 3.4.1 A revised relative depth loss

Chen et al. [17] define the loss term $\psi(i_k, j_k, r_k, z)$ as

$$\begin{cases} \ln\left(1+\exp(-z_{i_k}+z_{j_k})\right), & r_k \in \{>\} \\ \ln\left(1+\exp(z_{i_k}-z_{j_k})\right), & r_k \in \{<\} \\ (z_{i_k}-z_{j_k})^2, & r_k \in \{=\} \end{cases} \tag{3.2}$$

This definition encourages two depth values to be as different as possible if their ground truth ordinal relation is an inequality, or as similar as possible if their ground truth relation is equality. It works well if relative depth is the only form of supervision, as shown by Chen et al. [17], but it is problematic when used in conjunction with annotations of surface normals. The problem is that it encourages the difference of two unequal depth values to be infinitely large. This can potentially conflict with annotations of surface normals, which encourage the depth values to have a specific difference to form a specific surface orientation.

To address this issue we revise the loss term by introducing a margin $\tau > 0$ that stops the loss from decreasing if two depth values supposed to be unequal are already at least $\tau$ apart and if two equal depth values supposed to be equal are apart by no more than $\tau$:

$$\begin{cases} \ln\left(1+\exp(-\min(z_{i_k}-z_{j_k},\tau))\right), & r_k \in \{>\} \\ \ln\left(1+\exp(-\min(z_{j_k}-z_{i_k},\tau))\right), & r_k \in \{<\} \\ \max(\tau^2, |z_{i_k}-z_{j_k}|^2), & r_k \in \{=\}. \end{cases} \tag{3.3}$$

To make the loss term compatible with surface normals, we make another modification. We add a softplus transform to the network to enforce positive depth. This is needed because a negative depth means that the object is behind the camera and will cause issues in computing surface normals from the predicted depth.

### 3.4.2 Angle-based surface normal loss

We now consider how to define the loss term $\phi(p_l, n_l, z)$ in Eqn. 3.1 that compares the predicted depth map $z$ with a ground truth surface normal $n_l$ at location $p_l$.

The first approach we propose is to derive a surface normal $\nu(z)_{p_l}$ at the same location from the predicted depth map $z$ and compare the derived normal to the ground truth. Here $\nu$ is a function that maps a depth map to a map of surface normals, and $\nu(z)_{p_l}$ is the derived surface normal at location $p_l$. The loss term can now be defined as the angular difference between the derived normal

and the ground truth normal, expressed as a dot product of the two normals:

$$\phi_l(p_l, n_l, z) = - < n_l, \nu(z)_{p_l} > . \tag{3.4}$$

We call this formulation the *angle-based surface normal loss*.

To derive surface normals from depth, i.e. to implement the function $\nu$, we first back-project the pixels to 3D points in the camera coordinate system, assuming a pinhole camera model with a known focal length $f$. In particular, a pixel located at $(x, y)$ on the image plane with depth $z'$ is mapped to the 3D point $(xz'/f, yz'/f, z')$:

$$\beta : (x, y, z') \rightarrow (xz'/f, yz'/f, z') \tag{3.5}$$

We then compute the surface normal $\nu(z)_{xy}$ for a pixel located at $(x, y)$ using the cross product of the two vectors formed by its adjacent four neighbors (top to bottom, left to right):

$$\begin{aligned} \nu(z)_{xy} = [\beta(x-1, y, z_{x-1,y}) - \beta(x+1, y, z_{x+1,y})] \\ \otimes [\beta(x, y-1, z_{x,y-1}) - \beta(x, y+1, z_{x,y+1})], \end{aligned} \tag{3.6}$$

where $\otimes$ denotes cross product and $\beta$ is the back-projection function in Eqn. 3.5. Combining Eqn. 3.5, and Eqn. 3.4 gives a loss term $\phi(p_l, n_l, z)$ that is differentiable with respect to the predicted depth $z$ and can be easily incorporated into backpropagation.

### 3.4.3 Depth-based surface normal loss

The angle-based surface normal loss is natural, and a network trained with this loss in addition to relative depth annotations should predict better depth, as measured by the metric error (comparing the predict depth with ground truth depth in terms of absolute difference). In our experiments, however, we observe that this is not always the case, especially with a large training set. In particular, we observe that a network will predict a depth map that gives better surface normals, but the depth map itself does not improve in terms of metric error.

This leads us to make one theoretical observation. The observation is that when a surface normal is pointing sideways, a small change of the surface normal corresponds to a disproportionally large change in depth values for the neighobring pixels. In other words, metric depth error is very sensitive to the depth values in regions of steep slopes, but the angle-based loss does not reflect this sensitivity (Fig. 3.7). This could result in the phenomenon that a decrease in the angle-based loss does not corresponds to any notable improvement of metric depth error—the network is not focusing on the steep slopes, the places that would make the most difference in metric depth error.

Based on this observation we propose an alternative loss formulation, which we call *depth-*

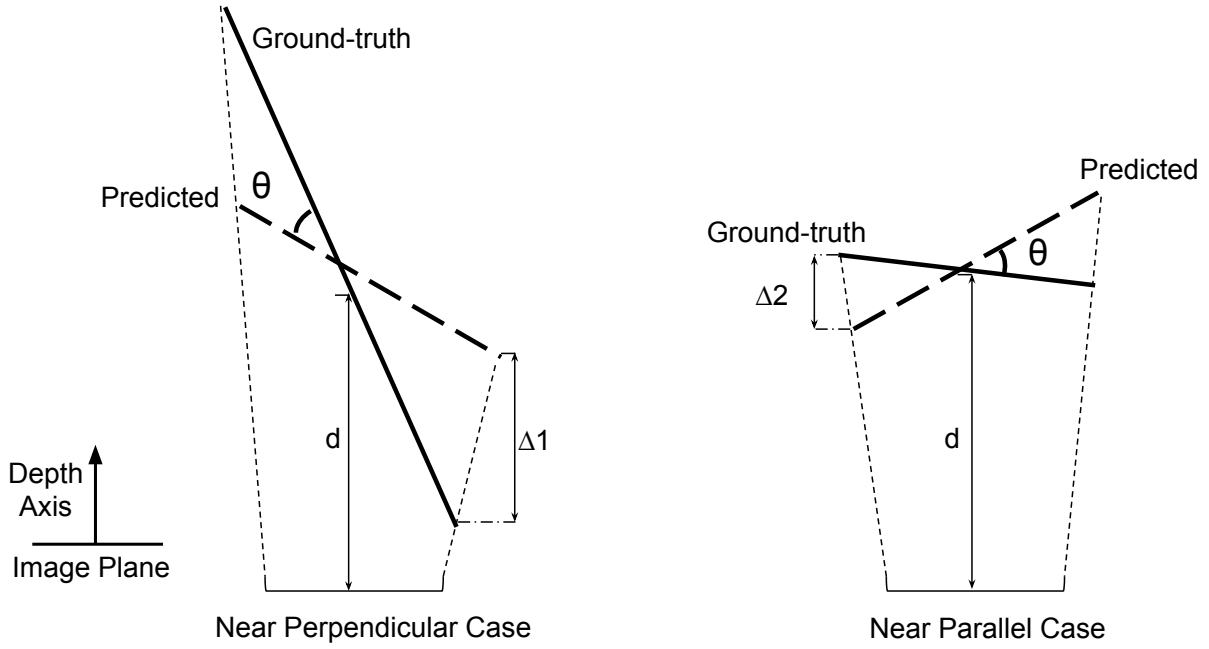



Figure 3.7: Two 3D planes (solid line) whose centers have the same distance $d$ to the image plane and whose projections occupy the same amount of area on an image. The predicted surface normals both deviate by $\theta$ from the ground-truth, but incur drastically different metric depth errors $\Delta 1$ and $\Delta 2$.

*based surface normal loss*. The idea is to take the predicted depth at a pixel and compute depth value of a neighbor using the ground truth normal. In other words, we compute the depth value the neighbor should take in order to be fully consistent with the ground truth normal. This "should-be" depth is compared with the actual predicted depth for the neighbor, and the difference becomes the penalty in the loss term. This loss is essentially converting a surface normal into the derivative of depth, and then compare it to the actual predicted derivative of depth. This depth-based loss is thus better aligned with metric depth error: surface normal annotations at steep slopes will play a bigger role in the loss.

Specifically, let $p^T$, $p^B$, $p^L$, $p^R$ be the top, bottom, left, right neighbors of pixel $p$. We first obtain the back projection $X^T$ of $p^T$ using the predicted depth $z_{p^T}$ (same as in Eqn. 3.5). Let $\Pi^T$ denote the plane that goes through $X^T$ and is oriented according to the ground truth normal $n_p$. By intersecting $\Pi^T$ with a ray that originates from the camera center and goes through the bottom neighbor $p^T$ in the image plane, we obtain the "should-be" depth value $\hat{z}_{p^B}$ for the bottom neighbor $p^B$. Similarly, we can obtain the "should-be" depth value for the top neighbor from the bottom neighbor ($\hat{z}_{p^T}$ from $z_{p^B}$), for the left neighbor from the right neighbor ($\hat{z}_{p^L}$ from $z_{p^R}$), and for the right neighbor from the left neighbor ($\hat{z}_{p^R}$ from $z_{p^L}$). Finally, the loss term is defined as the

difference between the "should-be" depth and the actual predicted depth for all neighbors.

$$\phi_l(p_l, n_l, z) = \sum_{i \in \{T,B,L,R\}} (\hat{z}_{p_l^i} - z_{p_l})^2 / (\hat{z}_{p_l^i} + z_{p_l})^2, \tag{3.7}$$

which is differentiable with respect to $z$. Note that the squared difference between the two depth values is normalized by their squared sum. This is for scale invariance; otherwise the network will minimize the loss mostly by shrinking the depth values with little regard to the normals.

### 3.4.4 Multiscale normals

In addition to introducing depth-based loss, we consider yet another strategy to address the issue of angle-based surface normal loss. The strategy is to collect surface normal annotations at multiple resolutions. That is, we can collect some surface normal annotations at lower resolutions. The rationale is that the steep slopes get smoothed out in lower resolutions and become less steep, which brings the angle-based loss more in line with metric depth error. To use the normals from lower resolutions, we add downsampling layers to the network to produce depth maps of lower resolutions, and add an angle-based loss at each additional resolution of the depth map.

## 3.5 Experiments on NYU Depth

We perform extensive experiments on NYU Depth [91]. The ground truth metric depth available in NYU Depth allows us to simulate and evaluate how adding surface normal annotations as indirect supervision can improve the prediction of metric depth, which is impossible for images in the wild, which do not have metric depth ground truth.

**Implementation details** For all our experiments on NYU Depth, we use the same network architecture proposed in [17]. The only difference is two modifications made to ensure that the loss term on relative depth will not encourage the predicted depth to deviate from the true metric depth, thus minimizing conflict with the loss term on surface normals. First, we add a softplus layer to ensure positive depth. Second, we take the log of the predicted depth before sending it to the relative depth loss in Eqn. 3.3. Taking the difference of the log depth is the same as taking the log of the depth ratio, which is more consistent with the relative depth annotations in NYU Depth [17, 123] because the ground truth ordinal depth relations are based on thresholding depth ratios rather than thresholding depth difference.

For relative depth "annotations" on NYU Depth, we use the same set as in [17]. For surface normal "annotations", we generate them from the ground-truth depth using Eq 3.6. Unless otherwise noted, in all our models trained with surface normals, we provide 5,000 surface normal annotations at random locations per image.

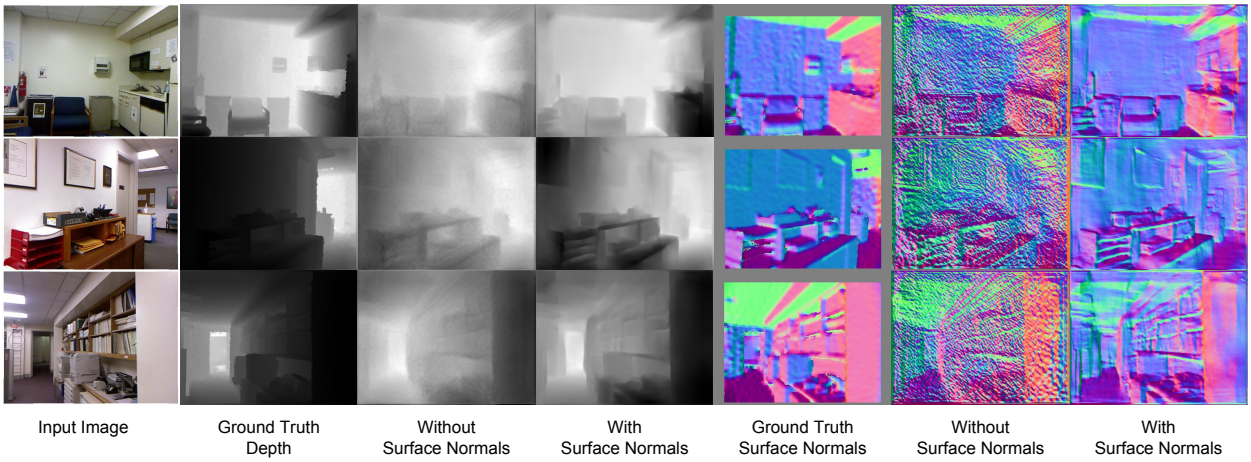



Figure 3.8: Qualitative results of the NYU test set. Here we show example outputs of the networks trained with or without surface normals on the NYU Subset.

**Main experiments** We compare 5 models: **(1)** a model trained with relative depth only (*d*); **(2)** a model trained with relative depth and surface normals using the angle-based loss (*d_n_al*); **(3)** same as (2) but using surface normals from multiple resolutions while keeping the total number of normal samples the same (*d_n_al_M*). **(4)** a model trained with relative depth and surface normals using depth-based loss (*d_n_dl*). **(5)** same as (4) but using surface normals from multiple resolutions while keeping the total number the same (*d_n_dl_M*).

As in prior work [17, 123], for each of the 5 models we train and evaluate on *NYU Subset*, a standard subset of 1449 images in NYU Depth, and *NYU Full*, the entire NYU Depth. Models trained on NYU Full are named with a *_F* suffix). Some qualitative results are shown in Fig. 3.8.

**Evaluating metric depth** Metric depth error measures the metric differences between the predicted depth map and the ground-truth depth map. Following prior work [17, 28, 123], we evaluate the root mean squared error (RMSE), the log RMSE, the log scale-invariant RMSE (log RMSE(s.inv)), the absolute relative difference (absrel) and the squared relative difference (sqrrel); their precise definitions can be found in [29]. Because single-image depth has scale ambiguity, before evaluation we normalize each predicted depth map such that it has the same mean and variance as those of the entire training set, as is done in [17].

However, such normalization is too crude in that it forces every predicted depth map to have the same mean and variance regardless of the input scene, which will unfairly penalize accurate predictions for scenes with a different mean and variance. We therefore propose a new error metric *Least-Square RMSE* (LS-RMSE) that better handles scale ambiguity in evaluation: for a predicted depth map $z$ and its ground-truth $z^*$ with pixels indexed by i, we compute the smallest possible

| Training Data | Method | RMSE | RMSE (log) | log RMSE (s.inv) | absrel | sqrrel | LS RMSE |
|---|---|---|---|---|---|---|---|
| NYU Subset | d | 1.12 | 0.39 | 0.26 | 0.36 | 0.45 | 0.64 |
| | d_n_al | 1.13 | 0.39 | 0.26 | 0.36 | 0.45 | 0.65 |
| | d_n_al_M | 1.11 | 0.39 | 0.25 | 0.36 | 0.44 | 0.59 |
| | **d_n_dl** | **1.11** | **0.39** | **0.25** | **0.35** | **0.44** | **0.58** |
| | d_n_dl_M | 1.11 | 0.39 | 0.25 | 0.36 | 0.45 | 0.59 |
| | Chen [17] | 1.12 | 0.39 | 0.26 | 0.36 | 0.46 | 0.65 |
| | Zoran [123] | 1.20 | 0.42 | - | 0.40 | 0.54 | - |
| NYU Full | d_F | 1.08 | 0.37 | 0.23 | 0.34 | 0.41 | 0.52 |
| | d_n_al_F | 1.09 | 0.38 | 0.24 | 0.34 | 0.42 | 0.55 |
| | d_n_al_F_M | 1.09 | 0.38 | 0.23 | 0.34 | 0.41 | 0.53 |
| | **d_n_dl_F** | **1.08** | **0.37** | **0.23** | **0.34** | **0.41** | **0.50** |
| | d_n_dl_F_M | 1.09 | 0.38 | 0.24 | 0.35 | 0.43 | 0.52 |
| | Chen_Full [17] | 1.09 | 0.38 | 0.24 | 0.34 | 0.42 | 0.58 |
| | Eigen(V)* [28] | 0.64 | 0.21 | 0.17 | 0.16 | 0.12 | 0.47 |
| | Chakrabarti* [14] | 0.64 | 0.21 | 0.17 | 0.15 | 0.12 | 0.47 |

Table 3.1: Metric depth error evaluated on the NYU Depth dataset. Models with a * suffix are trained on full metric depth.

sum of their squared differences under a global scaling and translation of the depth values:

$$\text{LS\_RMSE}(z, z^*) = \min_{a,b} \sum_i (az_i + b - z_i^*)^2. \tag{3.8}$$

Note that computing this error metric is the same as finding the least square solution to a system of linear equations, which has a well-known closed form solution.

Tab. 3.1 reports the results on metric depth error. We can see that our baseline model trained with relatived depth only matches or exceeds the metric depth error reported by Chen et al. [17]. We attribute this improvement to our revised relative depth loss (Eqn. 3.3), which does not encourage exaggerating depth differences once the ordering is correct.

On both NYU Subset and NYU Full, adding surface normals in training achieves significant improvement in metric depth quality, as reflected most notably in LS-RMSE. The improvement in metrics other than LS-RMSE is less significant, indicating a mismatch of depth scale. Among the models trained with surface normals, the one trained with the depth-based loss (*d_n_dl_F*) performs the best, as expected from our discussion in Sec. 3.4. On NYU Full, it outperforms the relative-depth-only baseline significantly on *LS RMSE*, approaching the models trained with full ground truth metric depth maps (Eigen(V) [28], Chakrabarti [14]).

The model trained with the angle-based normal loss yields no improvemet on NYU Subset and negative improvement on NYU Full, which can be explained by our theoretical observation that the angle-based loss is misaligned with the metric depth error. The misalignment is especially notable

on a bigger dataset, which is harder to fit and can cause the network to "give up" on the steep slopes, which account for very little in the angle-based normal loss. Using multiscale normals helps as expected, but it is not enough to overcome the misalignment on NYU Full to outperform the relative-depth-only baseline.

**Evaluating relative depth** We also evaluate a predicted depth map on ordinal error: disagreement with ground truth ordinal relations between selected locations. We use the same set of ground truth ordinal relations from [17], and report the same metrics: WKDR, the weighted disagreement rate between the predicted ordinal relations and the ground-truth ordinal relations, and its variants $\text{WKDR}^{=}$ (WKDR of pairs whose ground-truth order is =) and $\text{WKDR}^{\neq}$(WKDR of pairs whose ground-truth order is either $>$ or $<$). Following [17], we predict the ordinal relation of point A and B by thresholding on difference of the predicted depth.

The results on relative depth are shown in Tab. 3.2. First it is interesting to observe that our relative-depth-only baseline model is slightly worse than Chen et al. [17], which also trains with only relative depth. We attribute this difference to our revised relative depth loss (Eqn. 3.3)—the loss in Chen et al. [17] encourages exaggerating depth differences, which leads to better relative depth performance at the expense of metric accuracy, as reflected by Tab. 3.1.

Interestingly, adding normals improves ordinal error, but only from the angle-based normal loss, not from the depth-based normal loss. This is because depth-based normal loss places great emphasis on getting the exact steep slopes, but this does not make any difference to ordinal error as long as the sign of the slope is correct.

**Evaluating surface normals** We now evaluate the predicted depth in terms of surface normals derived from it. We use the same metrics as in [28]: the mean and median of angular difference with the ground-truth, and the percentages of predicted samples whose angular difference with the ground-truth are under a certain threshold. The ground truth normals for test are from NYU Depth toolkit [91], as is done in [106, 28]. We also evaluate the *derived* surface normals from other depth-estimation models, including **(1)** state-of-the-art depth estimation method of Eigen [28] and Chakrabarti [14]; **(2)** The original method of Chen et al. [17] augmented with a softplus layer to ensure positive depth but otherwise trained the same way with relative depth only (Chen* and Chen_Full*).

We report the results in Tab. 3.3. As expected, models trained with the angle-based normal loss perform better than any other models in terms of surface normals derived from depth, as the loss directly targes the normal error metric.

For reference, we also evaluate state of art methods that *directly predict* surface normals: Bansal [6], Eigen [28] and Wang [106]. Note that these models are trained on the full dense normal maps on NYU Full whereas our models are trained with only a sparse set of normals. Yet our best model ($d\_n\_al\_F$) outperforms Wang [106].

| Training Data | Method | WKDR | $\text{WKDR}^{=}$ | $\text{WKDR}^{\neq}$ |
|---|---|---|---|---|
| NYU Subset | d | 37.6% | 36.4% | 39.3% |
| | d_n_al | 36.5% | 35.5% | 37.9% |
| | **d_n_al_M** | **34.6%** | **33.4%** | **36.3%** |
| | d_n_dl | 38.7% | 36.9% | 40.5% |
| | d_n_dl_M | 39.0% | 37.7% | 40.5% |
| | Chen [17] | 35.6% | 36.1% | 36.5% |
| | Zoran [123] | 43.5% | 44.2% | 41.4% |
| NYU Full | d_F | 29.2% | 32.5% | 28.0% |
| | **d_n_al_F** | **27.6%** | **31.5%** | **26.6%** |
| | d_n_al_F_M | 27.9% | 32.2% | 26.6% |
| | d_n_dl_F | 30.9% | 31.7% | 31.4% |
| | d_n_dl_F_M | 35.5% | 38.9% | 34.6% |
| | Chen_Full [17] | 28.3% | 30.6% | 28.6% |
| | Eigen(V)* [28] | 34.0% | 43.3% | 29.6% |
| | Chakrabarti* [14] | 27.5% | 30.0% | 27.5% |

Table 3.2: Ordinal error evaluated on the NYU Depth dataset. Models with a * suffix are trained on full metric depth.

| Training Data | Method | Angle Distance | | % Within $t^\circ$ | | |
|---|---|---|---|---|---|---|
| | | Mean | Median | 11.25° | 22.5° | 30° |
| NYU Subset | d | 45.46 | 40.62 | 7.56 | 23.65 | 35.10 |
| | d_n_al | 37.53 | 31.93 | 13.04 | 34.38 | 47.39 |
| | **d_n_al_M** | **35.39** | **29.51** | **15.50** | **38.43** | **51.40** |
| | d_n_dl | 40.53 | 34.58 | 11.40 | 31.13 | 43.56 |
| | d_n_dl_M | 41.88 | 35.76 | 10.73 | 29.69 | 41.88 |
| | Chen* [17] | 50.68 | 44.96 | 4.16 | 16.77 | 28.21 |
| NYU Full | d_F | 29.45 | 22.71 | 22.31 | 50.71 | 63.65 |
| | **d_n_al_F** | **25.92** | **20.09** | **26.28** | **56.45** | **69.26** |
| | d_n_al_F_M | 26.50 | 20.42 | 26.41 | 55.47 | 68.09 |
| | d_n_dl_F | 30.85 | 24.51 | 24.51 | 46.93 | 60.31 |
| | d_n_dl_F_M | 37.63 | 31.58 | 13.41 | 34.97 | 47.97 |
| | Chen_Full* [17] | 30.35 | 24.37 | 18.64 | 46.80 | 61.42 |
| | Eigen(V) [28] | 35.97 | 28.34 | 17.67 | 41.12 | 53.49 |
| | Chakrabarti [14] | 29.80 | 20.43 | 31.34 | 54.90 | 64.57 |
| | Wang§ [106] | 28.8 | 17.9 | 35.2 | 57.1 | 65.5 |
| | Eigen(V)§ [28] | 22.89 | 16.26 | 38.23 | 63.30 | 73.18 |
| | Bansal§ [6] | **22.63** | **15.78** | **39.17** | **64.17** | **73.77** |

Table 3.3: Surface normal error evaluated on the NYU Depth dataset. The lower the better for Angle Distance metrics. The higher the better for the Percentage within $t^\circ$ metrics. Models with a § suffix directly predict surface normals.

| Crop | Method | RMSE | RMSE (log) | log RMSE (s.inv) | absrel | sqrrel | LS RMSE |
|---|---|---|---|---|---|---|---|
| Eigen | d | 7.61 | 2.11 | 1.94 | 0.39 | 3.16 | 5.99 |
| | d_n_al | 7.54 | 1.71 | 1.57 | 0.37 | 2.86 | 5.93 |
| | d_n_dl | 7.03 | 0.89 | 0.79 | 0.30 | 2.28 | 5.24 |
| | **Godard [35]** | **5.74** | **0.24** | **0.22** | **0.13** | **1.14** | **5.17** |
| Garg | d | 6.86 | 2.06 | 1.92 | 0.38 | 2.77 | 5.66 |
| | d_n_al | 6.75 | 1.56 | 1.45 | 0.34 | 2.45 | 5.57 |
| | d_n_dl | 6.17 | 0.83 | 0.76 | 0.28 | 1.88 | 4.84 |
| | **Godard [35]** | **5.21** | **0.22** | **0.20** | **0.11** | **0.89** | **4.73** |

Table 3.4: Metric depth error evaluated on the KITTI dataset.

**Discussion** Our expriements on NYU Depth show that surface normal annotations can help depth estimation in the absence of ground truth depth. We have proposed two different surface normal losses. Each has a different set of trade-offs and is appropriate in different applications. If metric fidelity is important, especially at depth discontinuities, then the depth-based loss is more appropriate. If surface orientation is important than the fidelity of depth discontinuities, then the angle-based loss is more appropriate.

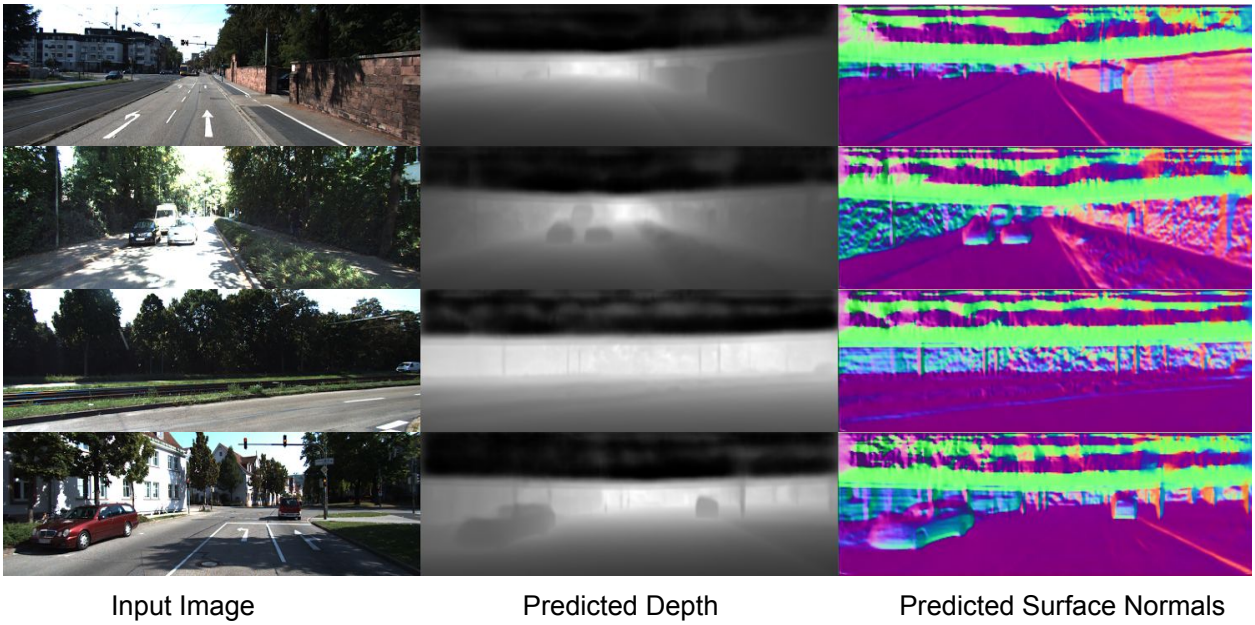



Figure 3.9: Qualitative results of the KITTI test set.

| Method | WKDR | $WKDR^{=}$ | $WKDR^{\neq}$ |
|---|---|---|---|
| d | 26.46% | 24.01% | 27.08% |
| **d_n_al** | **22.35%** | **20.61%** | **22.93%** |
| d_n_dl | 26.50% | 22.58% | 27.50% |
| Godard [35] | 25.84% | 26.17% | 26.21% |

Table 3.5: Ordinal error evaluated on the KITTI dataset.

## 3.6 Experiment on KITTI

For completeness, we provide experimental results on the KITTI dataset. Following [35], we evaluate our methods on two sub-regions of the KITTI test images (i.e., the *Garg_Crop* and *Eigen_Crop* as described in [35]), and use the test/train split of [29].

The relative depth annotations for both training and testing are generated in the same way as described in [123]. As the ground truth surface normals are not provided in the official KITTI dataset, we train on the surface normals generated by Eq.(6) of the paper, and only provide qualitative results of surface normal prediction on the test set. During training, we provide 5,000 surface normal annotations per image.

We test and compare these 3 models: **(1)** a model trained with relative depth only (*d*); **(2)** a model trained with relative depth and surface normals using the angle-based loss (*d_n_al*); **(3)** a model trained with relative depth and surface normals using depth-based loss (*d_n_dl*). We use the same network as used in the NYU experiment, with $\tau = ln(1.02)$ and $\lambda = 1$. The input to our network is a $128 \times 416$ image and the output is a depth map of the same size. Although ground truth depth values are only available on the lower part of the image, we feed the entire image into the network as is done in [29]. All the metric errors except the LS_RMSE are calculated by first normalizing the depth map to have the same mean and standard variation as the training set. However, some depth maps may contain negative depth value after normalization, and we replace those negative values with the minimum of the non-negative depth values of that depth map when calculating the RMSE (log) and log RMSE (s.inv) metric. For comparison, we also show the state-of-the-art depth-prediction results of Godard et al. [35], which exploits epipolar geometry constraints to train monocular depth-prediction networks (we show the results from their ***Ours resnet pp*** model, which is their best performing model).

We show the results in Tab. 3.4, 3.5. Some qualitative results are shown in Fig. 3.9. Models trained with surface normals (*d_n_al*, *d_n_dl*) consistently outperform the depth-only model (*d*) in both metric error and ordinal error. Training with depth-based loss yields the most significant improvement in metric error while the improvement in ordinal error is the most significant for the

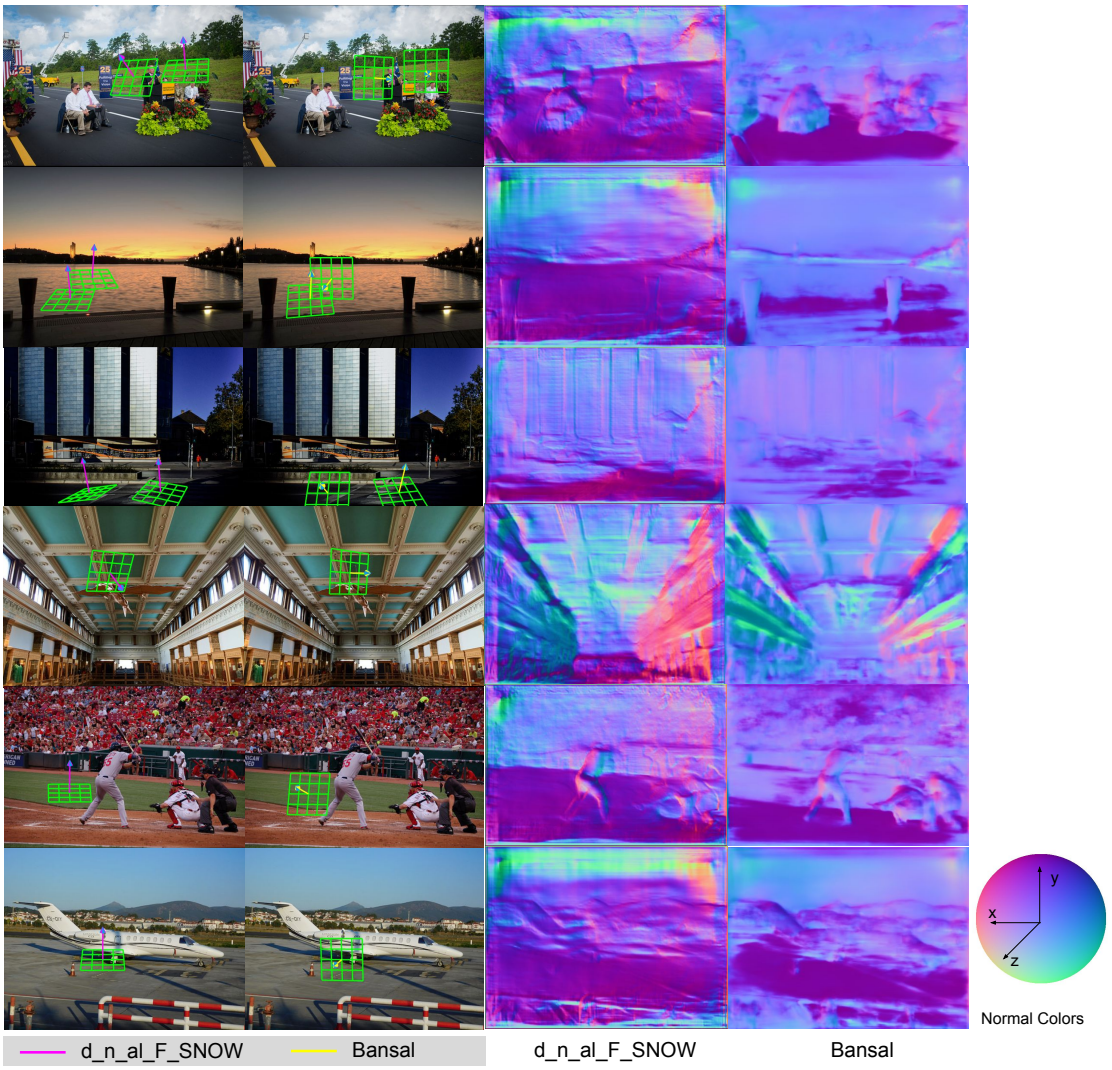



Figure 3.10: Normal maps produced by our model and Bansal [6]. Please view in color.

angle-based loss model. These results once again show that surface normals can help improve depth predictions in the absence of ground truth depth in training.

## 3.7 Experiments on SNOW

Since SNOW provides no ground truth of metric depth, it is infeasible to evaluate how training with surface normals helps predict metric depth. We thus evaluate surface normals as an indirect indicator of depth quality for images in the wild. We split SNOW into 10,256 test images and 49,805 training images.

We first evaluate the surface normals *derived* from depth prediction. Our baselines include state-of-the-art depth estimation methods Eigen [28] and FCRN [55], both trained with full metric

| | Model | Angle Distance | | Within $t^\circ$ | | |
|---|---|---|---|---|---|---|
| | | Mean | Median | 11.25° | 22.5° | 30° |
| Normals from Predicted Depth | d_n_al_F | 32.53 | 27.44 | 15.40 | 40.52 | 54.12 |
| | **d_n_al_F_SNOW** | **25.75** | **21.26** | **21.66** | **52.98** | **67.88** |
| | Chen_Full [17] | 35.16 | 30.26 | 13.70 | 36.56 | 49.56 |
| | Eigen(V) [28] | 48.71 | 46.15 | 6.35 | 18.91 | 28.45 |
| | FCRN [55] | 48.74 | 45.38 | 5.84 | 18.29 | 28.25 |
| Directly Predicted Normals | Ours_NYU§ | 31.96 | 26.03 | 18.16 | 43.72 | 56.03 |
| | **Ours_NYU_SNOW§** | **23.33** | **17.99** | **30.42** | **60.54** | **72.74** |
| | Eigen(V)§ [28] | 28.71 | 23.16 | 20.98 | 48.78 | 61.84 |
| | Bansal§ [6] | 27.85 | 22.25 | 23.41 | 50.54 | 64.09 |

Table 3.6: Surface normal error evaluated on SNOW. Models with a § suffix directly predict surface normals.

depth from NYU Full. We compare these baselines with the *d_n_al_F* network, our best performing model in terms of normal error. We also fine tune the *d_n_al_F* network on SNOW (*d_n_al_F_SNOW*).

We can see in Tab. 3.6 that our network trained only on NYU Full (*d_n_al_F*) already outperforms the baselines. Fine-tuning on SNOW yields a significant improvement.

SNOW also enables us to evaluate on methods that *directly predict* surface normals. We include four models: **(1)** state-of-the-art surface normal estimation methods of Bansal [6] and Eigen [28]; **(2)** Chen et al. [17]'s network trained to directly predict normals (*Ours_NYU§*); **(3)** *Ours_NYU§* fine-tuned on SNOW (*Ours_NYU_SNOW§*). We can see from Tab. 3.6 that fine-tuning on SNOW significantly improves surface normal prediction. Finally, Fig. 3.10 shows examples of qualitative improvement achieved by our network on images in the wild.

## 3.8 Summary

We have proposed two distinct approaches for using surface normal annotations to train a deep network that directly predicts per-pixel metric depth. We have also introduced a new dataset of crowdsourced surface normals for images in the wild (SNOW). Experiments show that surface normal annotations can advance depth estimation in the wild.

# CHAPTER 4

# OASIS: A Large-Scale Dataset for Single Image 3D in the Wild [1]

## 4.1 Introduction

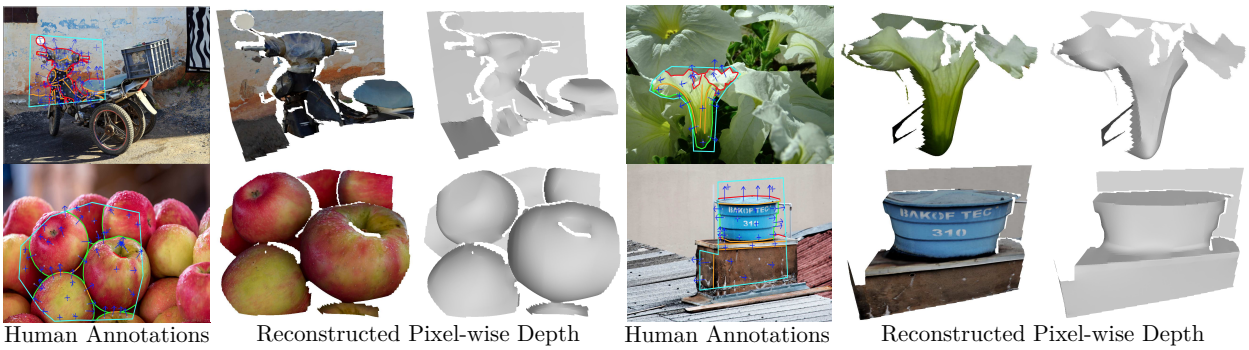



Figure 4.1: We introduce Open Annotations of Single-Image Surfaces (OASIS), a large-scale dataset of human annotations of 3D surfaces for 140,000 images in the wild. More examples in the supplementary material.

So far, we have discussed collecting relative depth and surface normals from the Internet through crowdsourcing. The data has been proven useful in advancing single-view 3D perception. However, these data are sparsely collected. On the other hand, human perception of 3D is dense — we can easily figure out the geometry of 3D surfaces and objects, and infer the connectivity among them. To fully realize the potential of human 3D perception and push the envelope of single-view 3D, this chapter explores collecting *pixel-wise* 3D ground truths in the wild.

Pixel-wise 3D ground truth is worthy of special attention as it has the potential to bring major advancement. Unlike object recognition, whose progress has been propelled by datasets like ImageNet [27] covering diverse object categories with high-quality labels, single-image 3D has lacked an ImageNet equivalent that covers diverse scenes with high-quality 3D ground truth. Existing datasets are restricted to either a narrow range of scenes [91, 26] or simplistic annotations such as

[1] This chapter is based on a joint work with Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng [19].

sparse relative depth pairs or surface normals [17, 20].

We introduce *Open Annotations of Single-Image Surfaces* (OASIS), a large-scale dataset for single-image 3D in the wild. It consists of human annotations that enable *pixel-wise* reconstruction of 3D surfaces for 140,000 randomly sampled Internet images. Fig. 4.1 shows the human annotations of example images along with the reconstructed surfaces.

A key feature of OASIS is its rich annotations of human 3D perception. Six types of 3D properties are annotated for each image: occlusion boundary (depth discontinuity), fold boundary (normal discontinuity), surface normal, relative depth, relative normal (orthogonal, parallel, or neither), and planarity (planar or not). These annotations together enable a reconstruction of pixelwise depth.

To construct OASIS, we created a UI for interactive 3D annotation. The UI allows a crowd worker to annotate the aforementioned 3D properties. It also provides a live, rotatable rendering of the resulting 3D surface reconstruction to help the crowd worker fine-tune their annotations.

It is worth noting that 100K images may not seem very large compared to millions of images in datasets like ImageNet. But the number of images can be a misleading metric. For OASIS, annotating one image takes 305 seconds on average. In contrast, verifying a single image-level label takes no more than a few seconds. Thus in terms of the total amount of human time, OASIS is already comparable to millions of image-level labels.

OASIS opens up new research opportunities on a wide range of single-image 3D tasks—depth estimation, surface normal estimation, boundary detection, and instance segmentation of planes—by providing in-the-wild ground truths either for the first time, or at a much larger scale than prior work. For depth estimation and surface normals, *pixelwise* ground truth is available for images in the wild for the first time—prior data in the wild provide only sparse annotations [17, 16]. For the detection of occlusion boundaries and folds, OASIS provides annotations at a scale 500 times larger than prior work—existing datasets [96, 46] have annotations for only about 200 images. For instance segmentation of planes, ground truth annotation is available for images in the wild for the first time.

To facilitate future research, we provide extensive statistics of the annotations in OASIS, and train and evaluate leading deep learning models on a variety of single-image tasks. Experiments show that there is a large room for performance improvement, pointing to ample research opportunities for designing new learning algorithms for single-image 3D. We expect OASIS to serve as a useful resource for 3D vision research.

## 4.2 Related Work

**3D Ground Truth from Depth-Sensors and Computer Graphics** Major 3D datatsets are either collected by sensors [91, 34, 83, 87, 26] or synthesized with Computer Graphics [12, 69, 95, 67, 78]. But due to the limitations of depth sensors and the lack of varied 3D assets for rendering, the diversity of scenes is quite limited. For example, sensor-based ground truth is mostly for indoor or driving scenes [91, 26, 69, 95, 34].

**3D Ground Truth from Multiview Reconstruction** Single-image 3D training data can also be obtained by applying classical Structure-from-Motion (SfM) algorithms on Internet images or videos [59, 108, 18]. However, classical SfM algorithms have many well known failure modes including scenes with moving objects and scenes with specular or textureless surfaces. In contrast, humans can annotate all types of scenes.

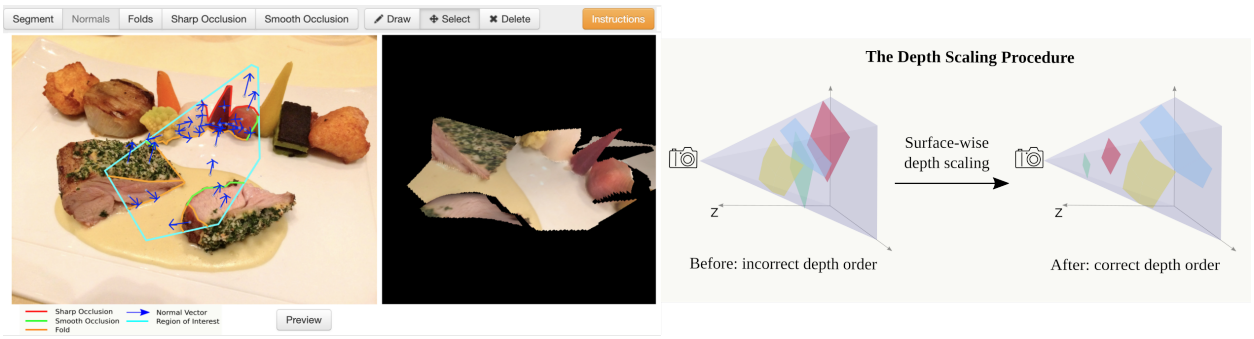
**3D Ground Truth from Human Annotations** Our work is connected to many previous works that crowdsource 3D annotations of Internet images. For example, prior work has crowdsourced annotations of relative depth [17] and surface normals [20] at sparse locations of an image (a single pair of relative depth and a single normal per image). Prior work has also aligned pre-existing 3D models to images [109, 97]. However, this approach has a drawback that not every shape can be perfectly aligned with available 3D models, whereas our approach can handle arbitrary geometry.

Our work is related to that of Karsch et al. [46], who reconstruct pixelwise depth from human annotations of boundaries, with the aid of a shape-from-shading algorithm [7]. Our approach is different in that we annotate not only boundaries but also surface normals, planarity, and relative normals, and our reconstruction method does not rely on automatic shape from shading, which is still unsolved and has many failure modes.

One of our inspirations is LabelMe3D [80], which annotated 3D planes attached to a common ground plane. Another is OpenSurfaces [10], which also annotated 3D planes. We differ from LabelMe3D and OpenSurfaces in that our annotations recover not only planes but also curved surfaces. Our dataset is also much larger, being $600\times$ the size of LabelMe3D and $5\times$ of OpenSurfaces in terms of the number of images annotated. It is also more diverse, because LabelMe3D and OpenSurface include only city or indoor scenes.

## 4.3 Crowdsourcing Human Annotations

We use random keywords to query and download Creative Commons Flickr images with a known focal length (extracted from the EXIF data). Each image is presented to a crowd worker for annotation through a custom UI as shown in Fig. 4.2 (a). The worker is asked to mask out a region that she wishes to work on with a polygon of her choice, with the requirement that the polygon




(a) (b)

Figure 4.2: **(a)** Our UI allows a user to annotate rich 3D properties and includes a preview window for interactive 3D visualization. **(b)** An illustration of the depth scaling procedure in our backend.

covers a pair of randomly pre-selected locations. She then works on the annotations and iteratively monitors the generated mesh (detailed in Sec 4.4) from an interactive preview window (Fig. 4.2 (a)).

**Occlusion Boundary and Fold** An occlusion boundary denotes locations of depth discontinuity, where the surface on one side is physically disconnected from the surface on the other side. When it is drawn, the worker also specifies which side of the occlusion is closer to the viewer, i.e. depth order of the surfaces on both sides of the occlusion. Workers need to distinguish between two kinds of occlusion boundaries. *Smooth occlusion* (green in Fig 4.2 (a)) is where the the closer surface smoothly curves away from the viewer, and the surface normals should be orthogonal to the occlusion line and parallel to the image plane, and pointing toward the further side. *Sharp occlusion* (red in Fig 4.2 (a)) has none of these constraints. On the other hand, *fold* denotes locations of surface normal discontinuity, where the surface geometry changes abruptly, but the surfaces on the two sides of the fold are still physically attached to each other (orange in Fig 4.2 (a)).

Occlusion boundaries segment a region into subregions, each of which is a *continuous surface* whose geometry can change abruptly but remains physically connected in 3D. Folds further segment a continuous surface into *smooth surfaces* where the geometry vary smoothly without discontinuity of surface normals.

**Surface Normal** The worker first specifies if a smooth surface is planar or curved. She annotates one normal at each planar surface which indicates the chap4:orientation of the plane. For each curved surface, she annotates normals at as many locations as she sees fit. A normal is visualized as a blue arrow originating from a green grid (Fig 4.3), rendered in perspective projection according to the known focal length. Such visualization helps workers perceive the normal in 3D [20]. To rotate and adjust the normal, the worker only needs to drag the mouse.

**Relative Normal** Finally, to annotate normals with higher accuracy, the worker specifies the *relative normal* between each pair of planar surfaces. She chooses between *Neither*, *Parallel* and

Figure 4.3: Surface normal annotation UI. The surface normal is visualized as a blue arrow originating from a green grid, rendered in perspective projection according to the known focal length.

*Orthogonal*. Surfaces pairs that are parallel or orthogonal to each other then have their normals adjusted automatically to reflect the relation.

**Interactive Previewing** While annotating, the worker can click a button to see a visualization of the 3D shape constructed from the current annotations (detailed later in Sec. 4.4). Workers can rotate or zoom to inspect the shape from different angles in a preview window (Fig 4.2 (a)). She keeps working on it until she is satisfied with the shape.

**Quality Control** Completing our 3D annotation task requires knowledge of relevant concepts. To ensure good quality of the dataset, we require each worker to complete a training course to learn concepts such as occlusions, folds and normals, and usage of the UI. She then needs to pass a qualification quiz before being allowed to work on our annotation task. Besides explicitly selecting qualified workers, we also set up a separate quality verification task on each collected mesh. In this task, a worker inspects the mesh to judge if it reflects the image well. Only meshes deemed high quality are accepted.

To improve our annotation throughput, we collected annotations from three sources: Amazon Mechanical Turk, which accounts for 11% of all annotations, and two data annotation companies that employ full-time annotators, who supplied the rest of the annotations. Some more collected annotations are shown in Fig 4.4.

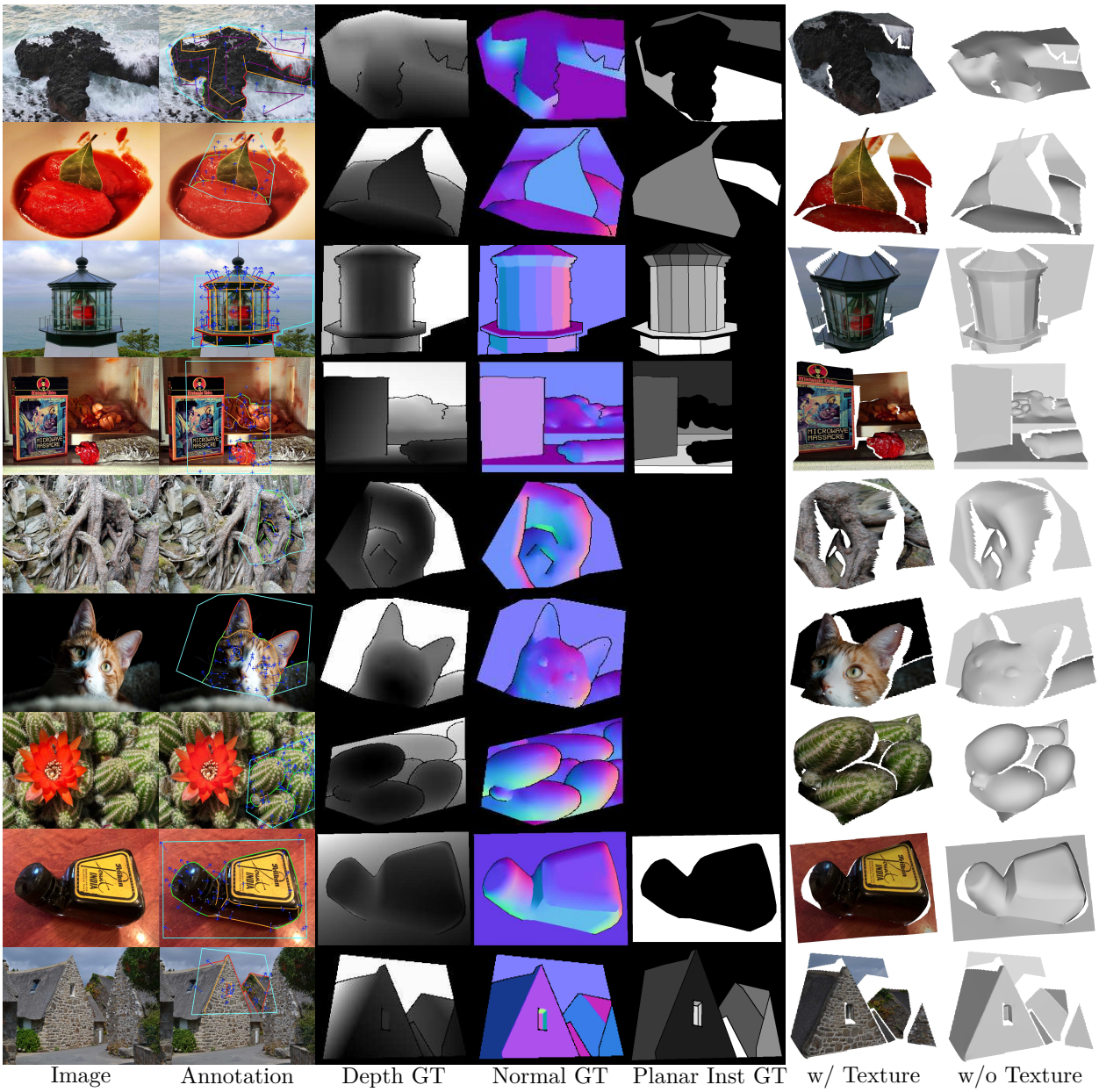



Figure 4.4: More human annotations from OASIS. Note that each planar instance has a different color.

## 4.4 From Human Annotations to Dense Depth

Because humans do not directly annotate the depth value of each pixel, we need to convert the human annotations to pixelwise depth in order to visualize the 3D surface.

**Generating Dense Surface Normals** We first describe how we generate dense surface normals from annotations. We assume the normals to be smoothly varying in the spatial domain, except across folds or occlusion boundaries where the normals change abruptly. Therefore, our system

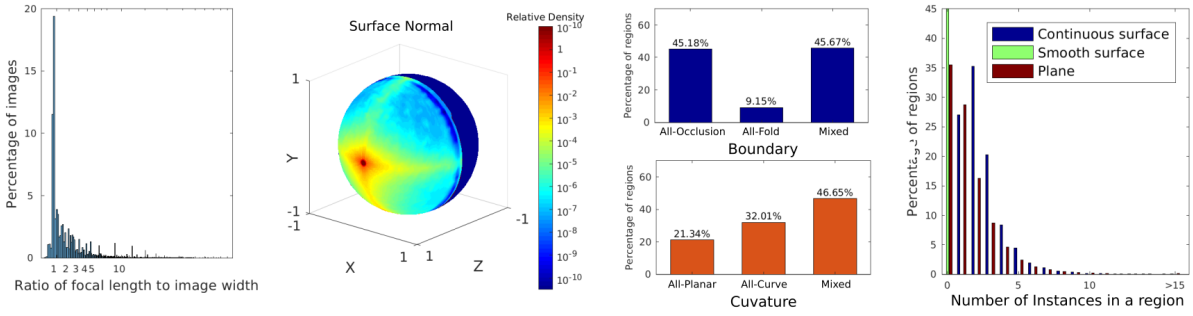



Figure 4.5: Statistics of OASIS. (a) The distribution of focal length (unit: relative length to the image width). (b) The distribution of surface normals. (c) Boundary: the ratio of regions containing only occlusion, only fold, and both. Curvature: the distribution of regions containing only planes, only curved surfaces, and both. (d) The frequency distribution of each surface type in a region.

propagates the known normals to the unknown ones by requiring the final normals to be smooth overall, but stops the propagation at fold and occlusion lines.

More concretely, let $N_p$ denote the normal at pixel $p$ on a normal map $N$, and $F$, $O$ denotes the pixels belong to the folds and occlusion boundaries. We have a set of known normals $\tilde{N}$ at locations $P_{known}$ from (1) surface normal annotations by workers, and (2) the pre-computed normals along the smooth occlusion boundaries as mentioned in Sec 4.3. Each pixel $p$ has four neighbors $\Phi(p)$. If $p$ is on an occlusion boundary, its neighbors on the closer side of this boundary are $\Gamma_O(p)$. If $p$ is on a fold line, only its neighbors $\Gamma_F(p)$ on one fixed random side of this line are considered. We solve for the optimal normal $N^*$ using LU factorization and then normalize it into unit norm:

$$\begin{aligned} N^* = \underset{N}{\operatorname{argmin}} \sum_{p \notin F \cup O} \sum_{\substack{q \in \Phi(p) \\ q \notin F \cup O}} |N_p - N_q|^2 + \\ \sum_{p \in O} \sum_{q \in \Gamma_O(p)} |N_p - N_q|^2 + \sum_{p \in F} \sum_{q \in \Gamma_F(p)} |N_p - N_q|^2 \end{aligned} \tag{4.1}$$

$$\text{s.t.} \quad N_p = \tilde{N}_p, \forall p \in P_{known} \tag{4.2}$$

**Generating Dense Depth** Our depth generation pipeline consists of two stages: First, from surface normals and focal length, we recover the depth of each *continuous surface* through integration [76]. Next, we adjust the depth order among these surfaces by performing surface-wise depth scaling (Fig. 4.2 (b)), i.e. each surface has its own scale factor.

Our design is motivated by this fact: in single-view depth recovery, depth within continuous surface can be recovered only up to an ambiguous scale; thus different surfaces may end up with different scales, leading to incorrect depth ordering between surfaces. But workers already decide

which side of an occlusion boundary is closer to the viewer. Based on such knowledge, we correct depth order by scaling the depth of each surface.

We now describe the details. Let $\mathbf{S}$ denotes the set of all continuous surface. From integration, we obtain the depth $Z_S$ of each $S \in \mathbf{S}$. We then solve for a scaling factor $X_S$ for each $S$, which is used in scaling depth $Z_S$. Let $\mathbf{O}$ denote the set of occlusion boundaries. Along $\mathbf{O}$, we densely sample a set of point pairs $\mathbf{B}$. Each pair $(p, q) \in \mathbf{B}$ has $p$ lying on the closer side of one of the occlusion boundaries $O_i \in \mathbf{O}$ and $q$ the further side. The continuous surface a pixel $p$ lies on is $S(p)$, and its depth is $Z_p$. The set of optimal scaling factors $\mathbf{X}^*$ is solved for as follows:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{S \in \mathbf{S}} X_S \tag{4.3}$$

$$\text{s.t.} \quad X_{S(p)} Z_p + \epsilon \leq X_{S(q)} Z_q, \forall (p, q) \in \mathbf{B} \tag{4.4}$$

$$X_S \geq \eta, \forall S \in \mathbf{S} \tag{4.5}$$

where $\epsilon > 0$ is a minimum separation between surfaces, and $\eta > 0$ is a minimum scale factor. Eq.(4.4) requires the surfaces to meet the depth order constraints specified by point pairs $(p, q) \in \mathbf{B}$ after scaling. Meanwhile, Eq.(4.3) constrains the value of $\mathbf{X}$ so that they do not increase indefinitely. After correcting the depth order, the final depth for surface $S$ is $X_S^* Z_S$. We normalize and reproject the final depth to 3D as point clouds, and generate 3D meshes for visualization.

| | NYU Depth [91] (depth mean: 2.471 m, depth std: 0.754 m) | | | Tanks & Temples [50] (depth mean: 4.309m, depth std: 3.059m) | | |
|---|---|---|---|---|---|---|
| | Human-Human | Human-Sensor | CNN-Sensor | Human-Human | Human-Sensor | CNN-Sensor |
| Depth (EDist) | 0.078m | 0.095m | 0.097m [55] | 0.194m | 0.213m | 0.402m [55] |
| Normals (MAE) | 13.13° | 17.82° | 14.19° [119] | 14.33° | 20.29° | 29.11° [119] |
| Post-Rotation Depth (EDist) | 0.037m | 0.048m | - | 0.082m | 0.080m | - |
| Depth Order (WKDR) | 5.68% | 8.67% | 11.90% | 9.28% | 10.80% | 32.13% |

Table 4.1: Depth and normal difference between different humans (Human-Human), between human and depth sensor (Human-Sensor), and between ConvNet and depth sensor (CNN-Sensor). The results are averaged over all human pairs.

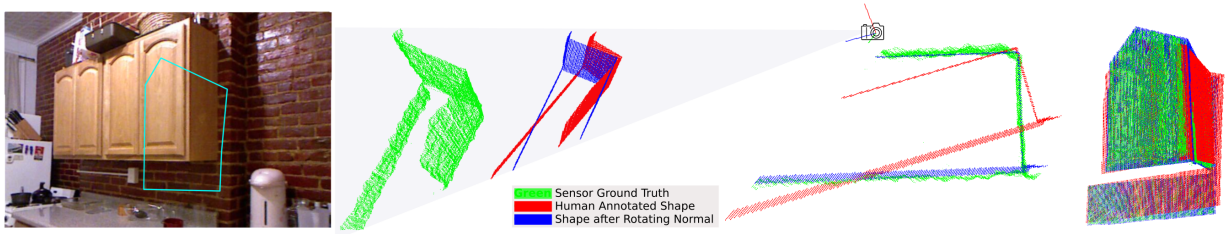



Figure 4.6: Humans estimate shape correctly but the absolute chap4:orientation can be slightly off, causing large depth error after perspective back-projection into 3D. Depth error drops significantly (from 0.07m to 0.01m) after a global rotation of normals.

## 4.5 Dataset Statistics

**Statistics of Surfaces** Fig. 4.5 plots various statistics of the 3D surfaces. Fig. 4.5 (a) plots the distribution of focal length. We see that focal lengths in OASIS vary greatly: they range from wide angle to telezoom, and are mostly $1\times$ to $10\times$ of the width of the image. Fig. 4.5 (b) visualizes the distribution of surface normals. We see that a substantial proportion of normals point directly towards the camera, suggesting that parallel-frontal surfaces frequently occur in natural scenes. Fig. 4.5 (c) presents region-wise statistics. We see that most regions (90%+) contain occlusion boundaries and close to half have both occlusion boundaries and folds (top). We also see that most regions (70%+) contain at least one curve surface (bottom). Fig. 4.5 (d) shows the histogram of the number of different kinds of surfaces in an annotated region. We see that most regions consist of multiple disconnected pieces and have non-trivial geometry in terms of continuity and smoothness.

**Annotation Quality** We study how accurate and consistent the annotations are. To this end, we randomly sample 50 images from NYU Depth [91] and 70 images from Tanks and Temples [50], and have 20 workers annotate each image. Tab. 4.1 reports the depth and normal difference between human annotations, between human annotations and sensor ground truth, and between predictions from state-of-the-art ConvNets and sensor ground truth. Depth difference is measured by the mean Euclidean distance (EDist) between corresponding points in two point clouds, after aligning one to the other through a global translation and scaling (surface-wise scaling for human annotations and CNN predictions). Normal difference is measured in Mean Angular Error (MAE). We see in Tab. 4.1 that human annotations are highly consistent with each other and with sensor ground truth, and are better than ConvNet predictions, especially when the ConvNet is not trained and tested on the same dataset.

We observe that humans often estimate the shape correctly, but the overall chap4:orientation can be slightly off, causing a large depth error against sensor ground truth (Fig. 4.6). This error can be particularly pronounced for planes close to orthogonal to the image plane. Thus we also compute the error after a rotational alignment with the sensor ground truth—we globally rotate the human annotated normals (up to 30 degrees) before generating the shape. After accounting for this global rotation of normals, human-sensor depth difference is further reduced by 47.96% (relative) for NYU and 62.44% (relative) for Tanks and Temples; a significant drop of normal error is also observed in human-human difference.

We also measure the qualitative aspect of human annotations by evaluating the WKDR metric [17], i.e. the percentage of point pairs with inconsistent depth ordering between query and reference depth. Depth pairs are sampled in the same way as [17]. Tab. 4.1 again shows that human annotations are qualitatively accurate and highly consistent with each other.

Finally, we evaluate the annotation quality separately for planar regions and curved regions.

Tab. 4.2 shows that humans are more consistent with each other when annotating curved regions than planar regions.

| | NYU Depth [91] | |
|---|---|---|
| | Human-Human | Human-Sensor |
| Planar Regions | 0.079m | 0.091m |
| Curved Regions | 0.077m | 0.102m |

Table 4.2: Depth difference between different humans (Human-Human) and between humans and depth sensors (Human-Sensor) in planar and curved regions. The results are averaged over all human pairs. The mean of depth in tested samples is 2.471 m, the standard deviation is 0.754 m.

It is worth noting that metric 3D accuracy is not required for many tasks such as navigation, object manipulation, and semantic scene understanding—humans do well without perfect metric accuracy. Therefore human perception of depth alone can be the gold standard for training and evaluating vision systems, regardless of its metric accuracy. As a result, our dataset would still be valuable even if it were less metrically accurate than it is currently.

**Comparison with Other Datasets** Tab. 4.3 compares OASIS and other datasets in terms of annotation types, size and diversity. OASIS provides a variety of in-the-wild 3D annotations either for the first time, or at a much larger scale than prior datasets.

## 4.6 Experiments

To facilitate future research, we use OASIS to train and evaluate leading deep learning models on a suite of single-image 3D tasks including depth estimation, normal estimation, boundary detection, plane segmentation. Qualitative results are shown in Fig. 4.7 (Qualitative predictions presented are produced as follows: Depth predictions are produced by a ResNetD [108] network trained on OASIS + ImageNet [27]. Surface normal predictions are produced by an Hourglass [20] network trained on OASIS alone. Occlusion boundary and fold predictions are produced by an Hourglass [17] network trained on OASIS alone. Planar instance segmentations are produced by a PlanarReconstruction [116] network trained on Scannet [26] + OASIS. More details on these models will be explained later in the chapter). A train-val-test split of 110K, 10K, 20K is used for all tasks.

For each task we estimate human performance to provide an upperbound accounting for the variance of human annotations. We randomly sample 100 images from the test set, and have each image re-annotated by 8 crowd workers. That is, each image now has "predictions" from 8 different humans. We evaluate each prediction and report the mean as the performance expected of an average human.

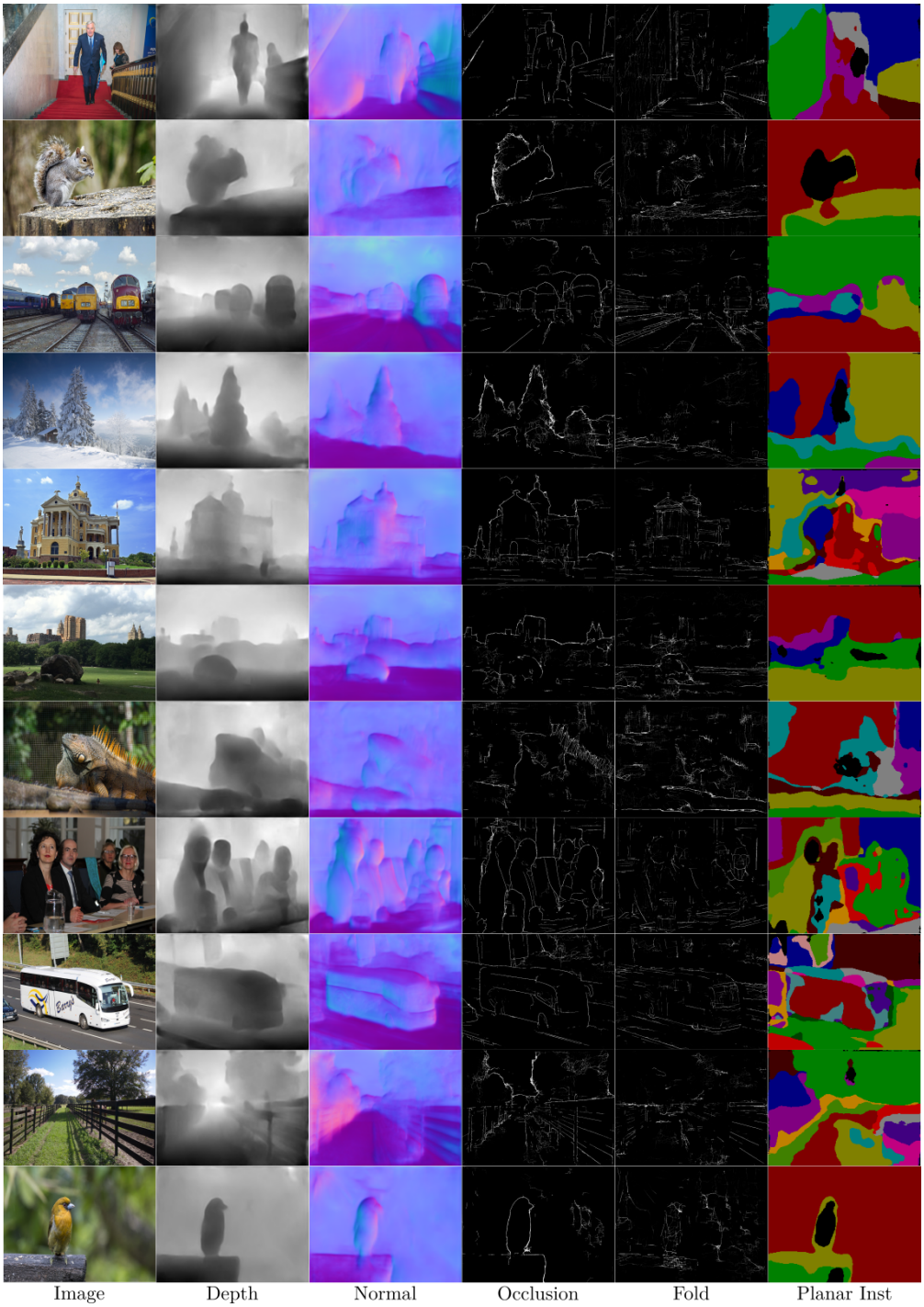



Figure 4.7: Qualitative outputs of the four tasks from representative models: (1) depth estimation, (2) normal estimation, (3) fold and occlusion boundary detection, and (4) planar instance segmentation.

| Dataset | In the Wild | Acquisition | Depth | Normals | Occlusion & Fold | Relative Normals | Planar Inst Seg | # Images |
|---|---|---|---|---|---|---|---|---|
| OASIS | ✓ | Human annotation | Metric (up to scale) | Dense | ✓ | ✓ | ✓ | 140K |
| NYU Depth V2 [91] | - | Kinect | Metric | Dense | - | - | - | 407K |
| KITTI [34] | - | LiDAR | Metric | - | - | - | - | 93K |
| DIW [17] | ✓ | Human annotation | Relative | - | - | - | - | 496K |
| SNOW [20] | ✓ | Human annotation | - | Sparse | - | - | - | 60K |
| MegaDepth [59] | ✓ | SfM | Metric (up to scale) | - | - | - | - | 130K |
| ReDWeb [108] | ✓ | Stereo | Metric (up to scale) | - | - | - | - | 3.6K |
| 3D Movie [56] | ✓ | Stereo | Metric (up to scale) | - | - | - | - | 75K |
| OpenSurfaces [10] | - | Human annotation | - | Dense | - | - | - | 25K |
| CMU Occlusion [96] | ✓ | Human annotation | - | - | Occlusion Only | - | - | 538 |

Table 4.3: Comparison between OASIS and other 3D datasets. *Metric (up to scale)* denotes that the depth is metrically accurate up to scale.

### 4.6.1 Depth Estimation

We first study single-view depth estimation. OASIS provides pixelwise *metric* depth in the wild. But as discussed in Sec 4.4, due to inherent single-image ambiguity, depth in OASIS is independently recovered within each continuous surface, after which the depth undergoes a surface-wise scaling to correct the depth order. The recovered depth is only accurate up to scaling within each continuous surface and ordering between continuous surfaces.

Given this, in OASIS we provide metric depth ground truths that is surface-wise accurate up to a scaling factor. This new form of depth necessitates new evaluation metrics and training losses.

**Depth Metric** The images in OASIS have varied focal lengths. This means that to evaluate depth estimation, we cannot simply use pixelwise difference between a predicted depth map and the ground truth map. This is because the predicted 3D shape depends greatly on the focal length—given the same depth values, decreasing the focal length will flatten the shape along the depth dimension. In practice, the focal length is often unknown for a test image. Thus, we require a depth estimator to predict a focal length along with depth. Because the predicted focal length may differ from the ground truth focal length, pixelwise depth difference is a poor indicator of how close the predicted 3D shape is to the ground truth.

A more reasonable metric is the Euclidean distance between the predicted and ground-truth 3D point cloud. Concretely, we backproject the predicted depth $Z$ to a 3D point cloud $\mathbf{P} = \{(X_p, Y_p, Z_p)\}$ using $f$ (the predicted focal length), and ground truth depth $Z^*$ to $\mathbf{P}^* = \{(X_p^*, Y_p^*, Z_p^*)\}$ using $f^*$ (the ground truth focal length). We then calculate the distance between $\mathbf{P}$ and $\mathbf{P}^*$.

The metric also needs to be invariant to surface-wise depth scaling and translation. Therefore we introduce a surface-wise scaling factor $\lambda_{S_i} \in \mathbf{\Lambda}$, and a surface-wise translation $\delta_{S_i} \in \mathbf{\Delta}$, to align each predicted surface $S_i \in \mathbf{S}$ in $\mathbf{P}$ to the ground truth point cloud $\mathbf{P}^*$ in a least square manner. The final metric, which we call Locally Scale-Invariant RMSE (LSIV_RMSE), is defined as:

$$\begin{aligned} LSIV\_RMSE(Z, Z^*) = \min_{\mathbf{\Lambda},\mathbf{\Delta}} \sum_p \Big(\frac{(X_p^*, Y_p^*, Z_p^*)}{\sigma(X^*)} \\ - \lambda_{S(p)}(X_p, Y_p, Z_p) - (0, 0, \delta_{S(p)})\Big)^2, \end{aligned} \tag{4.6}$$

where $S(p)$ denotes the surface a pixel $p$ is on. The ground truth point cloud $\mathbf{P}^*$ is normalized to a canonical scale by the standard deviation of its X coordinates $\sigma(X^*)$. Under this metric, as long as $\mathbf{P}$ is accurate up to scaling and translation, it will align perfectly with $\mathbf{P}^*$, and get 0 error.

Note that LSIV_RMSE ignore the ordering between two separate surfaces; it allows objects floating in the air to be arbitrarily scaled. This is typically not an issue because in most scenes there are not many objects floating in the air. But we nonetheless also measure the correctness of depth ordering. We report WKDR [17], which is the percentage of point pairs that have incorrect

depth order in the predicted depth. We evaluate on depth pairs sampled in the same way as [17], i.e. half are random pairs, half are from the same random horizontal lines.

**Models** We train and evaluate two leading depth estimation networks on OASIS: the Hourglass network [17], and ResNetD [108], a dense prediction network based on ResNet50. Each network predicts a metric depth map and a focal length, which are together used to backproject pixels to 3D points, which are compared against the ground truth to compute the LSIV_RMSE metric, which we optimize as the loss function during training. Note that we do not supervise on the predicted focal length.

We also evaluate leading pre-trained models that estimate single-image depth on OASIS, including FCRN [55] trained on ILSVRC [79] and NYU Depth [91], Hourglass [59] trained on MegaDepth [59], ResNetD [108] trained on a combination of datasets including ILSVRC [79], Depth in the Wild [17], ReDWeb [108] and YouTube3D [18]. For networks that do not produce a focal length, we use the validation set to find the best focal length that leads to the smallest LSIV_RMSE, and use this focal length for each test image. In addition, we also evaluate *plane*, a naive baseline that predicts a uniform depth map.

| Method | Training Data | LSIV_RMSE | WKDR |
|---|---|---|---|
| FCRN [55] | ImageNet [79] + NYU [91] | 0.67 | 39.95% |
| Hourglass [17, 59] | MegaDepth [59] | 0.67 | 38.37% |
| ResNetD [108, 18] | ImageNet [79] + YouTube3D [18]+ ReDWeb [108] + DIW [17] | 0.66 | 34.01% |
| ResNetD [108] | ImageNet [79] + OASIS | 0.37 | 32.62% |
| ResNetD [108] | OASIS | 0.47 | 39.73% |
| Hourglass [17] | OASIS | 0.45 | 39.01% |
| Plane | - | 0.67 | 100.00% |
| Human (Approx) | - | 0.24 | 19.04% |

Table 4.4: Depth estimation performance of different networks on OASIS (lower is better). For networks that do not produce a focal length, we use the best focal length leading to the smallest error.

Tab. 4.4 reports the results. In terms of metric depth, we see that networks trained on OASIS perform the best. This is expected because they are trained to predict a focal length and to directly optimize the LSIV_RMSE metric. It is noteworthy that ImageNet pretraining provides a significant benefit even for this purely geometrical task. Off-the-shelf models do not perform better than the naive baseline, probably because they were not trained on diverse enough scenes or were not trained to optimize metric depth error. In terms of relative depth, it is interesting to see that ResNetD trained on ImageNet and OASIS performs the best, even though the training loss does not enforce depth ordering. We also see that there is still a significant gap between human performance and machine performance. At the same time, the gap is not hopelessly large, indicating the effectiveness of a large training set.

| Method | Training Data | OASIS | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Angle Distance | | % Within $t^\circ$ | | | Relative Normal | |
| | | Mean | Median | 11.25° | 22.5° | 30° | $AUC_o$ | $AUC_p$ |
| Hourglass [20] | OASIS | 23.91 | 18.16 | 31.23 | 59.45 | 71.77 | 0. | 0.5786 |
| Hourglass [20] | SNOW [20] | 31.35 | 26.97 | 13.98 | 40.20 | 56.03 | 0. | 0.5016 |
| Hourglass [20] | NYU [91] | 35.32 | 29.21 | 14.23 | 37.72 | 51.31 | 0. | 0.5132 |
| PBRS [119] | NYU [91] | 38.29 | 33.16 | 11.59 | 32.14 | 45.00 | 0. | 0.5253 |
| Front_Facing | - | 31.79 | 24.80 | 27.52 | 46.61 | 56.80 | 0.5000 | 0.5000 |
| Human (Approx) | - | 17.27 | 12.92 | 44.36 | 76.16 | 85.24 | 0.8826 | 0.6514 |

Table 4.5: Surface normal estimation on OASIS.

| Method | Training Data | DIODE [103] | | | | ETH3D [87] | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Angle Distance | % Within $t^\circ$ | | | Angle Distance | % Within $t^\circ$ | | |
| | | Mean | 11.25° | 22.5° | 30° | Mean | 11.25° | 22.5° | 30° |
| Hourglass [20] | OASIS | 34.21 | 14.45 | 36.98 | 51.36 | 33.00 | 26.25 | 54.07 | 65.36 |
| Hourglass [20] | SNOW [20] | 40.10 | 8.29 | 27.20 | 40.67 | 45.71 | 10.69 | 31.16 | 43.16 |
| Hourglass [20] | NYU [91] | 42.23 | 10.97 | 29.76 | 41.35 | 41.84 | 21.94 | 44.05 | 53.81 |
| PBRS [119] | NYU [91] | 42.59 | 9.96 | 29.08 | 40.72 | 39.91 | 18.68 | 44.76 | 56.08 |
| Front_Facing | - | 47.76 | 5.62 | 18.70 | 28.05 | 58.97 | 11.84 | 23.75 | 30.19 |

Table 4.6: Cross-dataset generalization.

### 4.6.2 Surface Normal Estimation

We now turn to single-view surface normal estimation. We evaluate on absolute normal, i.e. the pixel-wise predicted normal values, and *relative normal*, i.e. the parallel and orthogonal relation predicted between planar surfaces.

**Absolute Normal Evaluation** We use standard metrics proposed in prior work [106]: the mean and median of angular error measured in degrees, and the percentage of pixels whose angular error is within $\gamma$ degrees.

We evaluate on OASIS four state-of-the-art networks that are trained to directly predict normals: (1) Hourglass [20] trained on OASIS, (2) Hourglass trained on the Surface Normal in the Wild (SNOW) dataset [20], (3) Hourglass trained on NYU Depth [91], and (4) PBRS, a normal estimation network by Zhang et al. [119] trained on NYU Depth [91]. We also include Front_Facing, a naive baseline predicting all normals to be orthogonal to the image plane.

Tab. 4.5 reports the results. As expected, the Hourglass network trained on OASIS performs the best. Although SNOW is also an in-the-wild dataset, the same network trained on it does not perform as well, but is still better than training on NYU. Notably, the human-machine gap appears fairly small numerically (17.27 versus 23.91 in mean angle error). However, we observe that the naive baseline can achieve 31.79; thus the dynamic range of this metric is small to start with, due to the natural distribution of normals in the wild. In addition, a close examination of the results suggests that these standard metrics of surface normals do not align well with perceptual quality. In natural images there can be large areas that dominate the metric but have uninteresting


Figure 4.8: Limitations of standard metrics: a deep network gets low mean angle error but important details are wrong.

geometry, such as a blank wall in the background. For example, in Fig. 4.8, a neural network gets the background correct, but largely misses the important details in the foreground. This opens up an interesting research question about developing new evaluation metrics.

**Relative Normal Evaluation** We also evaluate the predicted normals in terms of relative relations, specifically orthogonality and parallelism. Getting these relations correct is important because it can help find vanishing lines and perform self-calibration.

We first define a metric to evaluate relative normal. From the human annotations, we first sample an equal number of point pairs from surface pairs that are parallel, orthogonal, and neither. Given a predicted normal map, we look at the two normals at each point pair and measure the angle $\theta$ between them. We consider them orthogonal if $|cos(\theta - 90°)| < cos(\Theta_o)$, and parallel if $|cos(\theta)| > cos(\Theta_p)$, where $\Theta_o$, $\Theta_p$ are thresholds. We then plot the Precision-and-Recall curve for orthogonal by varying $\Theta_o$, and measure its Area Under Curve $\mathbf{AUC_o}$, using *neither* and *parallel* pairs as negative examples. Varying $\Theta_p$ and using *neither* and *orthogonal* as negative examples, we obtain $\mathbf{AUC_p}$ for parallel.

Tab. 4.5 reports results of relative normal evaluation. Notably, all methods perform similarly, and all perform very poorly compared to humans. This suggests that existing approaches to normal estimation have limitations in capturing orthogonality and parallelism, indicating the need for further research.

**Cross-Dataset Generalization** Next we study how networks trained on OASIS generalize to other datasets. Surface normal estimation is ideal for such evaluation because unlike depth, which is tricky to evaluate on a new dataset due to scale ambiguity and varying focal length, a normal estimation network can be directly evaluated on a new dataset without modification.

We train the same Hourglass network on OASIS, and NYU, and report their performance on two benchmarks not seen in training: DIODE [103] and ETH3D [87]. From Tab. 4.6 we see that training on NYU underperforms on all benchmarks, showing that networks trained on scene-specific datasets have difficulties generalizing to diverse scenes. Training on OASIS outperforms

on all benchmarks, demonstrating the effectiveness of diverse annotations.

### 4.6.3 Fold and Occlusion Boundary Detection

Occlusion and fold are both important 3D cues, as they tell us about physical connectivity and curvature: *Occlusion* delineates the boundary at which surfaces are physically disconnected to each other, while *Fold* is where geometry changes abruptly but the surfaces remain connected.

**Task** We investigate joint boundary detection and occlusion-versus-fold classification: deciding whether a pixel is a boundary (fold or occlusion) and if so, which kind it is. Prior work has explored similar topics: Hoiem et al. [41] and Stein et al. [96] handcraft edge or motion features to perform occlusion detection, but our task involves folds, not just occlusion lines.

| Metric \ Model | Edge: All Fold | Edge: All Occ | HED [112] | Hourglass [17] | Human (Approx) |
|---|---|---|---|---|---|
| ODS | 0.123 | 0.539 | 0.547 | 0.581 | 0.810 |
| OIS | 0.129 | 0.576 | 0.606 | 0.639 | 0.815 |
| AP | 0.02 | 0.44 | 0.488 | 0.530 | 0.642 |

Table 4.7: Boundary detection performance on OASIS.

**Evaluation Metric** We adopt metrics similar to standard ones used in edge detection [2, 112]: F-score by optimal threshold per image (OIS), by fixed threshold (ODS) and average precision (AP). For a boundary to be considered correct, it has to be labeled correctly as either occlusion or fold.

To perform joint detection of fold and occlusion, we adapt and train two networks on OASIS: Hourglass [17], and a state-of-the-art edge detection network HED [112]. The networks take in an image, and output two probabilities per pixel: $p_e$ is the probability of being an boundary pixel (occlusion or fold), and $p_f$ is the probability of being a fold pixel. Given a threshold $\tau$, pixels whose $p_e < \tau$ are neither fold nor occlusion. Pixels whose $p_e > \tau$ are fold if $p_F > 0.5$ and otherwise occlusion.

More specifically, the input to our evaluation pipeline consists of (1) the probability of each pixel being on edge (fold or occlusion) $p_e$, and (2) a label of each pixel being occlusion or fold. By thresholding on $p_e$, we first obtain an edge map $E_\tau$ at threshold $\tau$. We denote the occlusion pixels as $O$ and the fold pixels as $F$. We find the intersection $O \cap E_\tau$ and use the same protocol as [2] to compare it against the ground-truth occlusion $O^*$ and obtain true positive count $\text{TF}_o$, false positive count $\text{FP}_o$ and false negative count $\text{FN}_o$. We follow the same protocol to compare $F \cap E_\tau$ against ground-truth fold $F^*$ and obtain $\text{TF}_f$, $\text{FP}_f$ and $\text{FN}_f$. We then calculate the joint counts TF, FP and FN: TP=$\text{TF}_o$+$\text{TF}_f$, FP=$\text{FP}_o$+$\text{FP}_f$ and FN=$\text{FN}_o$+$\text{FN}_f$. We iterate through different $\tau$ to obtain the joint counts TF, FP and FN at each threshold to obtain the final ODS/OIS F-score and AP.

As baselines, we also investigate how a generic edge detector would perform on this task. We use HED network trained on BSDS dataset [2] to detect image edges, and classify the resulting edges to be either all occlusion (*Edge: All Occ*) or all fold (*Edge: All Fold*).

All results are reported on Tab 4.7. Hourglass outperforms HED when trained on OASIS, and significantly outperforms both the All-Fold and All-Occlusion baselines, but still underperforms humans by a large margin, suggesting that fold and occlusion boundary detection remains challenging in the wild.

### 4.6.4 Instance Segmentation of Planes

Our last task focuses on instance segmentation of planes in the wild. This task is important because planes often have special functional roles in a scene (e.g. supporting surfaces, walls). Prior work has explored instance segmentation of planes, but is limited to indoor or driving environments [64, 116, 63, 115]. Thanks to OASIS, we are able to present the first-ever evaluation of this task in the wild.

We follow the way prior work [64, 63, 116] performs this task: a network takes in an image, and produces instance masks of planes, along with an estimate of planar parameters that define each 3D plane. To measure performance, we report metrics used in instance segmentation literature [61]: the average precision (AP) computed and averaged across a range of overlap thresholds (ranges from 50% to 95% as in [61, 24]). A ground truth plane is considered correctly detected if it overlaps with one of the detected planes by more than the overlap threshold, and we penalize multiple detection as in [24]. We also report the AP at 50% overlap ($AP^{50\%}$) and 75% overlap ($AP^{75\%}$).

PlanarReconstruction by Yu et al. [116] is a state-of-the-art method for planar instance segmentation. We train PlanarReconstruction on three combinations of data: (1) ScanNet [26] only as done in [116], (2) OASIS only, and (3) ScanNet + OASIS. Tab. 4.8 compares their performance.

As expected, training on ScanNet alone performs the worse, because ScanNet only has indoor images. Training on OASIS leads to better performance. Leveraging both ScanNet and OASIS is the best overall. But even the best network significantly underperforms humans, suggesting ample space for improvement.

| Method | Training Data | AP | $AP^{50\%}$ | $AP^{75\%}$ |
|---|---|---|---|---|
| PlanarReconstruction [116] | ScanNet [26] | 0.076 | 0.161 | 0.064 |
| | OASIS | 0.125 | 0.249 | 0.110 |
| | ScanNet [26] + OASIS | 0.137 | 0.262 | 0.126 |
| Human (Approx) | - | 0.461 | 0.542 | 0.476 |

Table 4.8: Planar instance segmentation performance on OASIS.

## 4.7 Summary

We have presented OASIS, a dataset of rich human 3D annotations. We trained and evaluated leading models on a variety of single-image tasks. We expect OASIS to be a useful resource for 3D vision research.

# CHAPTER 5

# Learning Single-Image Depth from Internet Videos [1]

## 5.1 Introduction

The previous chapters discussed how we collect a large amount of 3D annotations by designing innovative UIs to crowdsource various 3D annotation tasks, with the goal to address the issue of lacking diverse data for single-view 3D in the wild. In particular, in Chapter 2, we crowdsourced human annotations of depth and constructed a dataset called "Depth-in-the-Wild (DIW)" that captures a broad range of scenes. This has been shown to be a feasible way to advance single-view depth estimation in the wild. One drawback of crowdsourcing, though, is that it requires a large amount of manual labor. What are other options that might be less dependent on manual labor? One way is to use synthetic data [12, 68, 78, 52], but it remains unclear how to automatically generate scenes that match the diversity of real-world images.

In this chapter, we explore a new approach that automatically collects single-view training data on natural in-the-wild images, without the need for crowdsourcing or computer graphics. The idea is to reconstruct 3D points from Internet videos using Structure-from-Motion (SfM), which matches feature points across video frames and infers depth using multiview geometry. The reconstructed 3D points can then be used to train single-view depth estimation. Because there is a virtually unlimited supply of Internet videos, this approach is especially attractive for generating a large amount of single-view training data.

However, to implement such an approach in practice, there remains a significant technical hurdle—despite great successes [1, 39, 85, 86, 70], existing SfM systems are still far from reliable when applied to arbitrary Internet videos. This is because SfM operates by matching features across video frames and reconstructing depth assuming a static scene, but feature matches are often unreliable and scenes often contain moving objects, both of which cause SfM to produce erroneous 3D reconstructions. That is, if we simply apply an off-the-shelf SfM system to arbitrary Internet videos, the resulting single-view training data will have poor quality.

[1] This chapter is based on a joint work with Shengyi Qian, and Jia Deng [18].

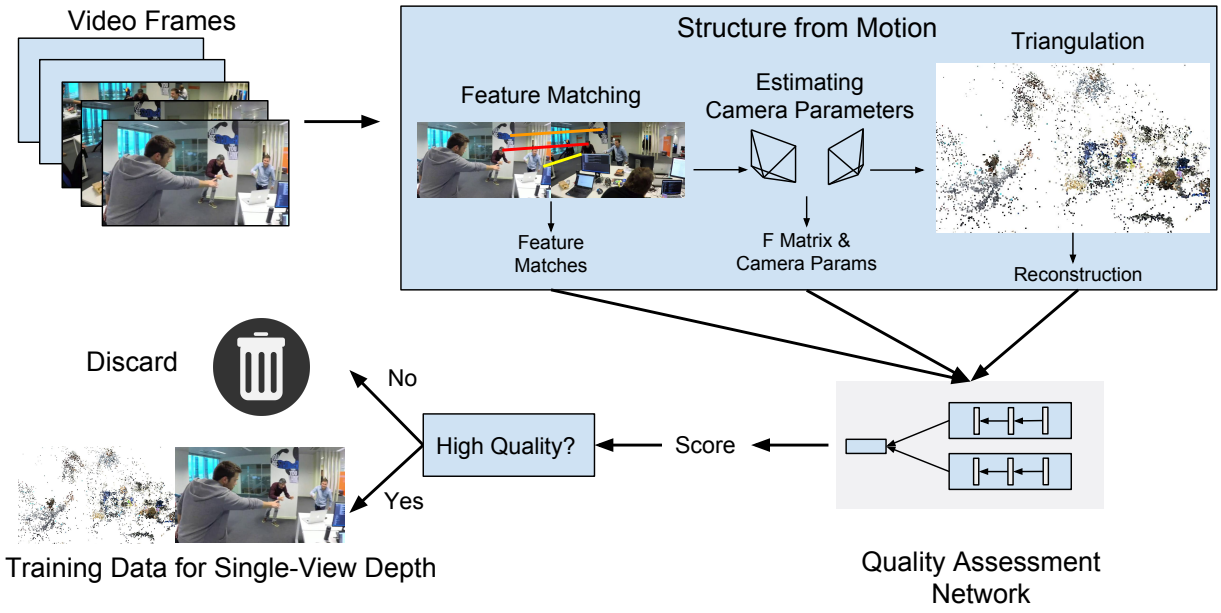



Figure 5.1: An overview of our data collection method. Given an arbitrary video, we follow standard steps of structure-from-motion: extracting feature points and matching them across frames, estimating the camera parameters, and performing triangulation to obtain a reconstruction. A Quality Assessment Network (QANet) examines the operation of the SfM pipeline and assigns a score to the reconstruction. If the score is above a certain threshold, this reconstruction is deemed of high quality, and we use it as single-view depth training data. Otherwise, the reconstruction is discarded.

To address this issue, we propose to train a deep network to automatically assess the quality of a SfM reconstruction. The network predicts a quality score of a SfM construction by examining the operation of the entire SfM pipeline—the input, the final output, along with intermediate outputs generated inside the pipeline. We call this network a *Quality Assessment Network (QANet)*. Using a QANet, we filter out unreliable reconstructions and obtain high-quality single-view training data. Fig. 5.1 illustrates our data collection method.

It is worth noting that because Internet videos are virtually unlimited, it is sufficient for a QANet to be able to reliably identify a small proportion of high-quality reconstructions. In other words, high precision is necessary but high recall is not. This means that training a QANet will not be hopelessly difficult because we do not need to detect *every* good reconstruction, only *some* good reconstructions.

We experiment using Internet videos in the wild. Our experiments show that with QANet integrated with SfM, we can collect high-quality single-view training data from unlabeled videos, and such training data can supplement existing data to significantly improve the performance of single-image depth estimation.

Using our proposed method, we constructed a new dataset called YouTube3D, which consists of 795K in-the-wild images, each associated with depth annotations generated from SfM reconstructions filtered by a QANet. We show that as a standalone training set for in-the-wild depth estimation, YouTube3D is superior to existing datasets constructed with human annotation. YouTube3D also outperforms MegaDepth [59], a recent datatset automatically collected through SfM on Internet images. In addition, we show that as a supplement to existing RGB-D data, YouTube3D advances the state-of-the-art of single-image depth estimation in the wild.

Our contributions are two fold: (1) we propose a new method to automatically collect high-quality training data for single-view depth by integrating SfM and a quality assessment network; (2) using this method we construct YouTube3D, a large-scale dataset that advances the state of the art of single-view depth estimation in the wild.

## 5.2 Related Work

### 5.2.1 RGB-D from depth sensors

A large amount of RGB-D data from depth sensors has played a key role in driving recent research on single-image depth estimation [33, 91, 15, 26, 87]. But due to the limitations of depth sensors and the manual effort involved in data collection, these datasets lack the diversity needed for arbitrary real world scenes. For example, KITTI [33] consists mainly of road scenes; NYU Depth [91], ScanNet [26] and Matterport3D [15] consist of only indoor scenes. Our work seeks to address this drawback by focusing on diverse images in the wild.

### 5.2.2 RGB-D from computer graphics

RGB-D from computer graphics is an attractive option because the depth will be of high quality and it is easy to generate a large amount. Indeed, synthetic data has been used in computer vision with much success [36, 101, 68, 99, 12, 30, 23, 107, 77]. In particular, SUNCG [94] has been shown to improve single-view surface normal estimation on natural indoor images from the NYU Depth dataset [118]. However, the diversity of synthetic data is limited by the availability of 3D "assets", i.e. shapes, materials, layouts, etc., and it remains difficult to automatically compose diverse scenes representative of the real world.

### 5.2.3 RGB-D from crowdsourcing

Crowdsourcing depth annotations [17, 20] has recently received increasing attention. It's appealing because it can be applied to a truly diverse set of in-the-wild images. Chen et al. [17] crowdsourced annotations of relative depth and constructed Depth in the Wild (DIW), a large-scale

dataset for single-view depth in the wild. The main drawback of crowdsourcing is, obviously, the cost of manual labor, and our work attempts to mitigate or avoid this cost through an automatic method.

### 5.2.4 RGB-D from multiview geometry

When multiple images of the same scene are available, depth can be reconstructed through multiview geometry. Prior work has exploited this fact to collect RGB-D data. Xian et al. [108] perform stereopsis on stereo images, i.e. pairs of images taken by two calibrated cameras, to collect a dataset called "ReDWeb". Li et al. [59] perform SfM on unordered collections of online images of the same scenes to collect a dataset called "MegaDepth".

Our work differs from prior work in two ways. First, we use a new source of RGB data—monocular videos—which likely offer better availability and diversity—stereo images have limited availability because they must be taken by stereo cameras. Multiple images of the same scene tend to be biased toward well-known sites frequented by tourists.

Second, our method of quality assessment is new. Both prior works performed some form of quality assessment, but neither used learning. Xian et al. [108] manually remove some poor reconstructions; Li et al. [59] use handcrafted criteria based on semantic segmentation. In contrast, our quality assessment network can learn criteria and patterns beyond those that are easy to handcraft.

### 5.2.5 Predicting failure

Our work is also related to prior work on predicting failures for vision systems [117, 25, 11, 5]. For example, Zhang et al. [117] predict failure for a variety of vision tasks based solely on the input. Daftry et al. [25] predict failures in an autonomous navigation system directly from the input video stream. Our method is different in that we predict failure in a SfM system to filter reconstructions, based not on the input images but on the outputs of the SfM system.

## 5.3 Approach

Our method consists of two main steps: SfM followed by quality assessment, as illustrated by Fig. 5.1. SfM produces candidate 3D reconstructions, which are then filtered by a QANet before we use them to generate single-view training data.

### 5.3.1 Structure from Motion

The SfM component of our method is standard. We first detect and match features across frames. We then estimate the fundamental matrix and perform triangulation to produce 3D points.

It is worth noting that SfM produces only a sparse reconstruction. Although we can generate a dense point cloud by a subsequent step of multiview stereopsis, we choose to forgo it, because stereopsis in unconstrained settings tends to contain a large amount of error, especially in the presence of low-texture surfaces or moving objects.

Our SfM component also involves a couple minor modifications compared to a standard full-fledged SfM system. First, we only perform two-view reconstruction. This is to simplify the task of quality assessment—the quality assessment network only needs to examine two input images as opposed to many. Second, we do not perform bundle adjustment [38], because we observe that with unknown focal length of Internet videos (we assume a centered principal point and focal length is the only unknown intrinsic parameter), it often leads to poor results. This is because bundle adjustment is sensitive to initialization, and tends to converge to an incorrect local minimum if the initialization of focal length is not already close to correct. Instead, we search a range of focal lengths and pick the one that leads to the smallest reprojection error after triangulation. This approach does not get stuck in local minima, and is justified by the fact that focal length can be uniquely determined when it is the only unknown intrinsic parameter of a fixed camera across two views [74].

### 5.3.2 Quality Assessment Network (QANet)

The task of a quality assessment network is to identify good SfM reconstructions and filter out bad ones. In this section we discuss important design decisions including the input, output, architecture, and training of a QANet.

**Input to QANet** The input to a QANet should include a variety of cues from the operation of a SfM pipeline on a particular input. Recall that we consider only two-view reconstruction; thus the input to SfM is only two video frames.

We consider cues associated with the entire reconstruction (reconstruction-wise cues) as well as those associated with each reconstructed 3D point (point-wise cues). Our reconstruction-wise cues include the inferred focal length and the average reprojection error. Our point-wise cues include the 2D coordinates of a feature match, the Sampson distance of a feature match under the recovered fundamental matrix, and the angle between the two rays connecting the reconstructed 3D point and the camera centers.

Note that we do not use any information from the pixel values. The QANet only has access to geometrical information of the matched featues. This is to allow better generalization by preventing overfitting to image content.

Also note that in a SfM pipeline RANSAC is typically used to handle outliers. That is, multiple reconstructions are attempted on random subsets of the feature matches. Here we apply the QANet

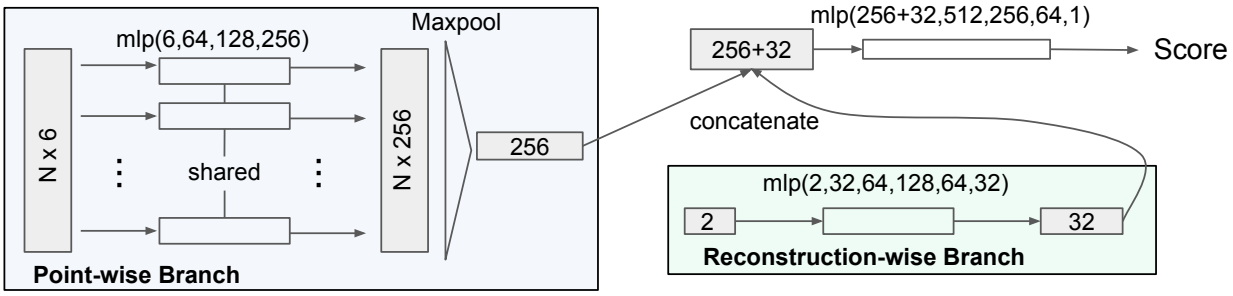



Figure 5.2: Architecture of the Quality Assessment Network (QANet).

only to the best subset free from outliers.

**Output of QANet** The output of a QANet is a quality score for the entire reconstruction, i.e. a sparse point cloud. Ideally, this score should correspond to a similarity metric between two point clouds, the reconstructed one and the ground truth.

There are many possible choices of the similarity metric, with different levels of invariance and robustness (e.g. invariance to scale, and robustness to deformation and outliers). Which one to use should be application dependent and is not the main concern of this work. And it is sufficient to note that our method is general and not tied to a particular similarity metric.

**QANet architecture** Fig. 5.2 illustrates the architecture of our QANet. It consists of two branches. The *reconstruction-wise branch* processes the reconstruction-wise cues (the focal length and overall reprojection error). The *point-wise branch* processes features associated with each reconstructed point. The outputs from the two branches are then concatenated and fed into multiple fully connected layers to produce a quality score.

Point-wise cues need a separate branch because they involve an unordered set of feature vectors with a variable size. To be invariant to the number and ordering of the vectors, we employ an architecture similar to that of PointNet [75]. In this architecture, each vector is independently processed by shared subnetwork and the results are max-pooled at the end.

**QANet training** To train a QANet, a straightforward approach is to use a regression loss that minimizes the difference between the predicted quality score and the ground truth score—the similarity between the reconstructed 3D point cloud and the ground truth.

However, using a regression loss makes learning harder than necessary. In fact, the absolute value of the score matters much less than the ordering of the score, because when we use a QANet for filtering, we remove all reconstructions with scores below a threshold, which can be chosen by cross-validation. In other words, the network just needs to tell that one construction is better

than another, but does not need to quantify the exact degree. Moreover, the precision of top-ranked reconstructions is much more important than the rest, and should be given more emphasis in the loss.

This observation motivates us to use a ranking loss. Let $s_1$ be the "ground truth quality score" (i.e. similarity to the ground truth reconstruction) of a reconstruction in the training set. Let $s'_1$ be its predicted quality score by the QANet. Similarly, let $s_2$ be the ground truth quality of another reconstruction, and let $s'_2$ be the predicted quality score. We define a ranking loss $h(s'_1, s'_2, s_1, s_2)$ on this pair of reconstructions:

$$h(s'_1, s'_2, s_1, s_2) = \begin{cases} \ln\left(1 + \exp(s'_2 - s'_1)\right), & \text{if } s_1 > s_2 \\ \ln\left(1 + \exp(s'_1 - s'_2)\right), & \text{if } s_1 < s_2 \end{cases} \tag{5.1}$$

This loss imposes a penalty if the score ordering of the pair is incorrect. When applied to all possible pairs, it generates a very large total penalty if a bad reconstruction is ranked top, because many pairs will have the wrong ordering. Obviously, in practice we cannot afford to train with all possible pairs. Instead, we uniformly sample random pairs whose difference in ground truth quality scores are larger than some threshold.

## 5.4 Experiments

**Relative depth** One implementation question we have left open in the previous sections is the choice of the "ground truth" quality score for the QANet. Specifically, to train an actual QANet, we need a similarity metric that compares a reconstructed point cloud with the ground truth point cloud (the clouds have the same number of points and known correspondence).

In our experiments we define the similarity metric based on relative depth. We consider all pairs of points in the reconstructed cloud, and calculate the percentage of pairs that have the same depth ordering as the ground truth. Note that depth ordering is view dependent, and because our SfM component performs two-view reconstruction, we take the average from both views.

Our choice of relative depth as the quality measure is motivated by two reasons. First, relative depth is more robust to outliers. Unlike metrics based on metric difference such as RMSE, with relative depth a single outlier point will not be able to dominate the error. Second, relative depth has been used as a standard evaluation metric for depth prediction in the wild [17, 58, 108, 114], partly because it would be difficult to obtain ground truth for arbitrary Internet images except to use humans, which are good at annotating relative depth but not metric depth.

Another implementation question is how to train a single-view depth network with the single-view data generated by our method, i.e. 3D points from SfM filtered by the QANet. Here we opt to also derive relative depth from the 3D points. In other words, the final form of our automatically

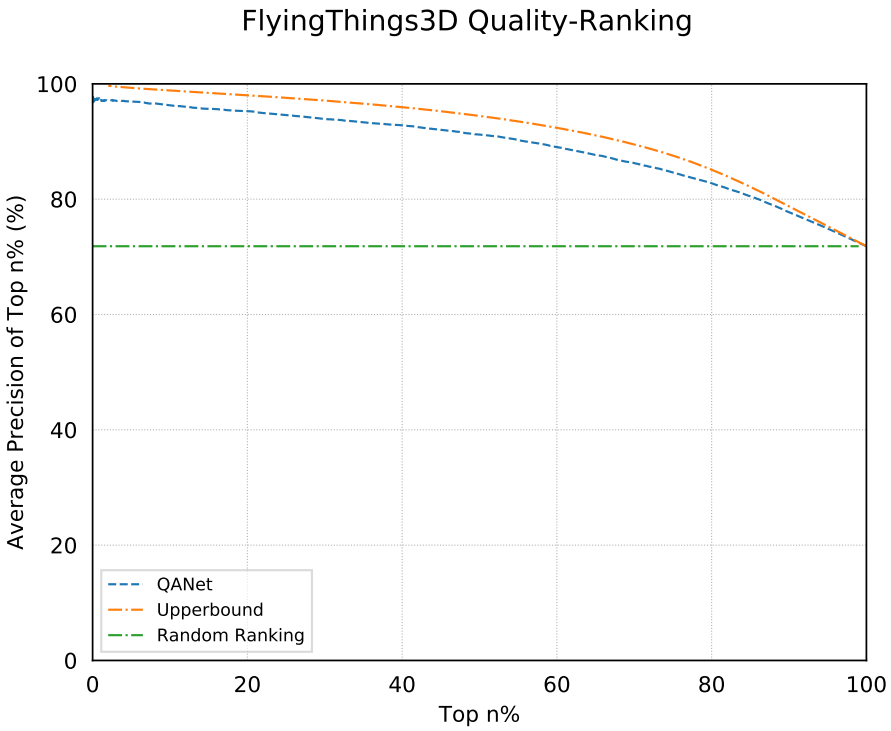



Figure 5.3: The quality-ranking curve on the FlyingThings3D dataset.

collected training data is a set of video frames, each associated with a set of 2D points with their "ground truth" depth ordering.

One advantage of using relative depth as training data is that it is scale-invariant and sidesteps the issue of scale ambiguity in our SfM reconstructions. In addition, prior work [17] has shown that relative depth can serve as a good source of supervision even when the goal is to predict dense metric depth. Last but not least, using relative depth allows us to compare our automatically collected data with prior work such as MegaDepth [59], which also generates training data in the form of relative depth.

### 5.4.1 Evaluating QANet

We first evaluate whether the QANet, as a standalone component, can be successfully trained to identify high-quality reconstructions.

We train the QANet using a combination of existing RGB-D video datasets: NYU Depth [91], FlyingThings3D [68], and SceneNet [69]. We use the RGB videos to produce SfM reconstructions and use the depth maps to compute the ground truth quality score for each reconstruction.

We measure the performance of our QANet by plotting a quality-ranking curve—the Y-axis is the average ground-truth quality (i.e. percentage of correct releative depth orderings) of the top $n\%$

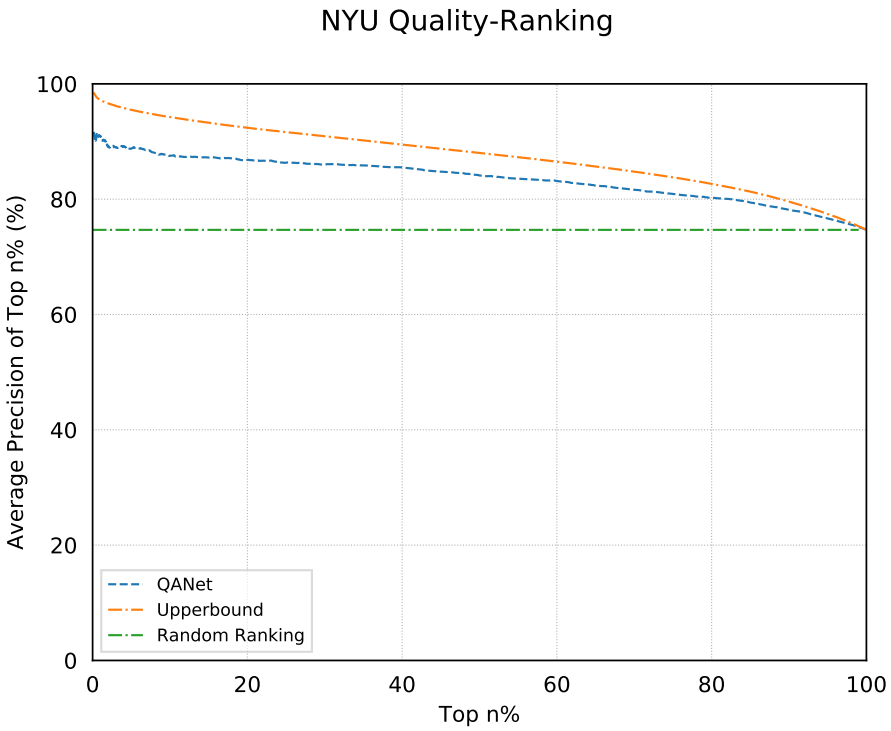



Figure 5.4: The quality-ranking curve on the NYU dataset.

reconstructions ranked by QANet, and the X-axis is the number $n$. At the same $n$, a better QANet would have a better average quality.

We test our QANet on the test splits of FlyingThings3D and NYU Depth. The results are shown in Fig. 5.3 and Fig. 5.4. In both figures, we provide an *Upperbound* curve from a perfect ranking of the reconstructions, and a *Random Ranking* curve from a random ranking of the reconstructions.

From Fig. 5.3 and Fig. 5.4 we see that our QANet can successfully rank reconstructions by quality. On FlyingThings3D, the average quality of unfiltered (or randomly ranked) reconstructions is 71.41%, whereas the top 20% reconstructions ranked by QANet have an average quality of 95.26%. On NYU Depth, the numbers are 75.09% versus 86.80%.

In addition, we see that the QANet curve is quite close to the upperbound curve. On FlyingThings3D, the AUC (area under curve) of the upperbound curve is 91.28%, and the AUC of QANet is 89.02%. On NYU Depth, the numbers are 87.49% and 83.56%.

**Ablative Studies** We next study the contributions of different cues to quality assessment. We train five ablated versions of QANet by (1) removing 2D coordinate feature (-2D); (2) removing Sampson distance feature (-Sam); (3) removing angle feature (-Ang); (4) removing focal length (-Focal); (5) removing reprojection error (-RepErr).

| QANet Variants | AUC | |
|---|---|---|
| | NYU | FlyingThings3D |
| -2D | 80.53% | 85.34% |
| -Sam | 83.20% | 88.66% |
| -Ang | 82.09% | 85.00% |
| -Focal | 82.54% | 88.37% |
| -RepErr | 83.37% | 88.50% |
| **Full** | **83.56%** | **89.02%** |
| Upperbound | 87.49% | 91.28% |
| Random Ranking | 75.09% | 71.41% |

Table 5.1: AUC (area under curve) for different ablated versions of the QANet.

Figure 5.5: Examples of automatically collected relative depth annotations in YouTube3D. The relative depth pairs are visualized as two connected points, with red point being closer than the blue point. These relative depth annotations are mostly correct.

We compare their performances in terms of AUC with the full QANet in Tab. 5.1. They all underperform the full QANet, indicating that all cues contribute to successful quality assessment.

### 5.4.2 Evaluating the full method

We now turn to evaluating our full data collection method. To this end, we need a way to compare our dataset with those collected by alternative methods.

Note that it is insufficient to compare datasets using the accuracy of the ground truth labels, because the datasets may have different numbers of images, different images, or different annotations on the same images (e.g. different pairs of points for relative depth). A dataset may have less accurate labels, but may still end up more useful due to other reasons such as better diversity or more informative annotations.

Instead, we compare datasets by their usefulness for training. In our case, a dataset is better if it trains a better deep network for single-view depth estimation. Given a dataset of relative depth, we use the method of Chen et al. [17] to train a image-to-depth network by imposing a ranking loss on the output depth values to encourage agreement with the ground truth orderings. We measure

the performance of the trained network by the weighted human disagreement rate (*WHDR*) [17], i.e. the percentage of incorrectly ordered point pairs.

**YouTube3D** We crawled 0.9 million YouTube videos using random keywords. Pairs of frames are randomly sampled and selected if feature matches exist between them. We apply our method to these pairs and obtain 2 million filtered reconstructions spanning 121,054 videos. From these reconstructions we construct a dataset called *YouTube3D*, which consists of 795,066 images, with an average of 281 relative depth pairs per image. Example images and annotations of YouTube3D are shown in Fig. 5.5.

As a baseline, we construct another dataset called $YT_{UF}$. It is built from all reconstructions that are used in constructing YouTube3D but without applying the QANet filtering. Note that $\text{YT}_{UF}$ is a superset of YouTube3D, and contains 3.5M images.

**Colmap** Our implementation of SfM is adapted from Colmap [85], a state-of-the-art SfM system. We use the same feature matches generated by Colmap, and modified the remaining steps as described in Sec. 5.3.1. In our experiments, we also include the original unmodified Colmap system as a baseline. To generate relative depth from the sparse point clouds given by Colmap, we randomly sample point pairs and project them into different views.

We run Colmap on the same set of features and matches as used in constructing YouTube3D and $\text{YT}_{UF}$, obtaining 647,143 reconstructions that span 486,768 videos. From them we construct a dataset called $YT_{Col}$. It contains 3M images, with an average of 4,755 relative depth pairs per image.

**Depth-in-the-Wild (DIW)** We use the Depth-in-the-Wild (DIW) dataset [17] to evaluate the performance of a single-view depth network. DIW consists of Internet images that cover diverse types of scenes. It has 74,000 test and 420,000 train images; each image has human annotated relative depth for one pair of points. In addition to using the test split of DIW for evaluation, we also use its training split as a standalone training set.

**Evaluation as standalone dataset** We evaluate YouTube3D as a standalone dataset and compare it with other datasets. That is, we train a single-view depth network from scratch using each dataset and measure the performance on DIW. To directly compare with existing results in the literature, we use the same hourglass network that has been used in a number of prior works [17, 59].

Tab. 5.2 compares the DIW performance of a hourglass network trained on YouTube3D against those trained on three other datasets: MegaDepth [58], NYU Depth [91], and the training split of DIW [17]. The results are shown in Tab. 5.2. We see that YouTube3D not only outperforms

| Training Sets | WHDR |
|---|---|
| NYU | 31.31% [17] |
| DIW | 22.14% [17] |
| MegaDepth | 22.97% [59] |
| $YT_{Col}$ | 34.47% |
| $YT_{UF}$ | 25.11% |
| QA_train | 31.77% |
| NYU + QA_train | 31.22% |
| **YouTube3D** | **19.01%** |

Table 5.2: Error rate on the DIW test set by the Hourglass Network [17] trained on different standalone datasets.

NYU Depth, which was acquired with depth sensors, but also MegaDepth, another high-quality depth dataset collected via SfM. Most notably, even though the evaluation is on DIW, YouTube3D outperforms the training split of DIW, showing that our automatic data collection method is a viable substitute for manual annotation.

Tab. 5.2 also compares YouTube3D against $YT_{UF}$ (YouTube3D without QANet filtering) and $YT_{Col}$ (off-the-shelf SfM). We see that YouTube3D outperforms the unfiltered set $YT_{UF}$ by a large margin, even though $YT_{UF}$ is a much larger superset of YouTube3D. This underscores the effectiveness of QANet filtering. Moreover, YouTube3D outperforms $YT_{Col}$ by an even larger margin, indicating our method is much better than a direct application of off-the-shelf state-of-the-art SfM to Internet videos. Notably, $YT_{UF}$ already outperforms $YT_{Col}$ significantly. This is a result of our modifications described in Sec. 5.3.1: (1) we require the estimate of the fundamental matrix to have zero outliers during RANSAC; (2) we replace bundle adjustment with a grid-search of focal length.

Fig. 5.6 shows a qualitative comparison of depth estimation by networks trained with different datasets. We can see that training on YouTube3D generally produces better results than others, especially compared to $YT_{Col}$ and NYU.

We also include a comparison between YouTube3D and *QA_train*, the data used to train QANet. This is to answer the question whether a naive use of this extra data—using it directly to train a single-view depth network—would give the same advantage enjoyed by YouTube3D, rendering our method unnecessary. We see in Tab. 5.2 that training single-view depth directly from QA_train is much worse than YouTube3D (31.77% vs. 19.01%), showing that QA_train itself is a not a good training set for mapping pixels to depth. In addition, adding QA_train to NYU Depth (NYU + QA_train in Tab. 5.2) barely improves the performance of NYU Depth alone. This shows that a naive use of this extra data will not result in the improvement achievable by our method. It also shows that QANet generalizes well to images in the wild, even when trained on data that is


Figure 5.6: Qualitative results on the DIW test set by the Hourglass Network [17] trained with different datasets. Column names denote the datasets used for training.

quite different in terms of pixel content. It is worth noting that this result should not be surprising, because QANet does not use pixel values to assess quality and only uses the geometry of the feature matches.

| Network | Training Sets | WHDR |
|---|---|---|
| Hourglass [17] | NYU + DIW | 14.39% [17] |
| | **NYU + DIW + YouTube3D** | **13.50%** |
| EncDecResNet [108] | ImageNet + ReDWeb | 14.33% |
| | ImageNet + ReDWeb + DIW | 11.37% |
| EncDecResNet (Our Impl of [108]) | ImageNet + ReDWeb | 16.31% |
| | ImageNet + YouTube3D | 16.21% |
| | ImageNet + ReDWeb + DIW | 12.03% |
| | **ImageNet + ReDWeb + DIW + YouTube3D** | **10.59%** |

Table 5.3: Error rate on the DIW test set by networks trained with and without YouTube3D as supplement.

**Evaluation as supplemental dataset** We evaluate YouTube3D as supplemental data. Prior works have demonstrated state-of-the-art performance on DIW by combining multiple sources of training data [17, 108]. We investigate whether adding YouTube3D as additional data would improve state-of-the-art systems.

We first add YouTube3D to NYU + DIW, the combined training set used by Chen et al. [17] to train the first state-of-art system for single-view depth in the wild. We train the same hourglass network used in [17]. Results in Tab. 5.3 show that with the addition of YouTube3D, the network is able to achieve a significant improvement.

We next evaluate whether YouTube3D can improve the best existing result on DIW, achieved by an encoder-decoder network based on ResNet50 [108] (which we will refer to as an EncDecResNet subsequently). The network is trained on a combination of ImageNet, DIW, and ReDWeb, a relative depth dataset collected by performing stereopsis on stereo images with manual removal of poor-quality reconstructions. Tab. 5.3 summarizes our results, which we elaborate below.

We implement our own version of the EncDecResNet used in [108], because there is no public code available as of writing. As a validation of our implementation, we train the network on ImageNet and ReDWeb, and achieve an error rate of $16.31\%$, which is slightly worse than but sufficiently close to the $14.33\%$ reported in [108][2]. This discrepancy is likely because certain details (e.g. the exact number of channels at each layer) are different in our implementation because they are not available in their paper.

As an aside, we train the same EncDecResNet on ImageNet and YouTube3D, which gives an error rate of $16.21\%$, which is comparable with the $16.31\%$ given by ImageNet and ReDWeb. This suggests that YouTube3D is as useful as ReDWeb. This is noteworthy because unlike ReDWeb, YouTube3D is not restricted to stereo images and does not involve any manual filtering. Note that it is not meaningful to compare with the $14.33\%$ reported in [108]—to compare two training

[2]All results in [108] are with ImageNet.

datasets we need to train the exact same network, but the $14.33\%$ is likely from a slightly different network due to the unavailability of some details in [108].

Finally, we train an EncDecResNet on the combination of ImageNet, DIW, and ReDWeb, which has produced the current state of the art on DIW in [108]. With our own implementation we achieve an error rate of $12.03\%$, slightly worse than the $11.37\%$ reported in [108]. Adding YouTube3D to the mix, we achieve an error rate of $10.59\%$, a new state of the art performance on DIW (see Fig. 5.7 for example depth estimates). This result demonstrates the effectiveness of YouTube3D as supplemental single-view training data.

**Discussion** The above results suggest that our proposed method can generate high-quality training data for single-view depth in the wild. Such results are significant, because our dataset is gathered by a *completely automatic* method, while datasets like DIW [17] and ReDWeb [108] are constrained by manual labor and/or the availability of stereo images. Our automatic method can be readily applied to a much larger set of Internet videos and thus has potential to advance the state of the art of single-view depth even more significantly.

## 5.5 Summary

In this chapter we propose a fully automatic and scalable method for collecting training data for single-view depth from Internet videos. Our method performs SfM and uses a Quality Assessment Network to find high-quality reconstructions, which are used to produce single-view depth ground truths. We apply the proposed method on YouTube videos and construct a single-view depth dataset called YouTube3D. We show that YouTube3D is useful both as a standalone and as a supplemental dataset in training depth predictors. With it, we obtain state-of-the-art results on single-view depth estimation in the wild.

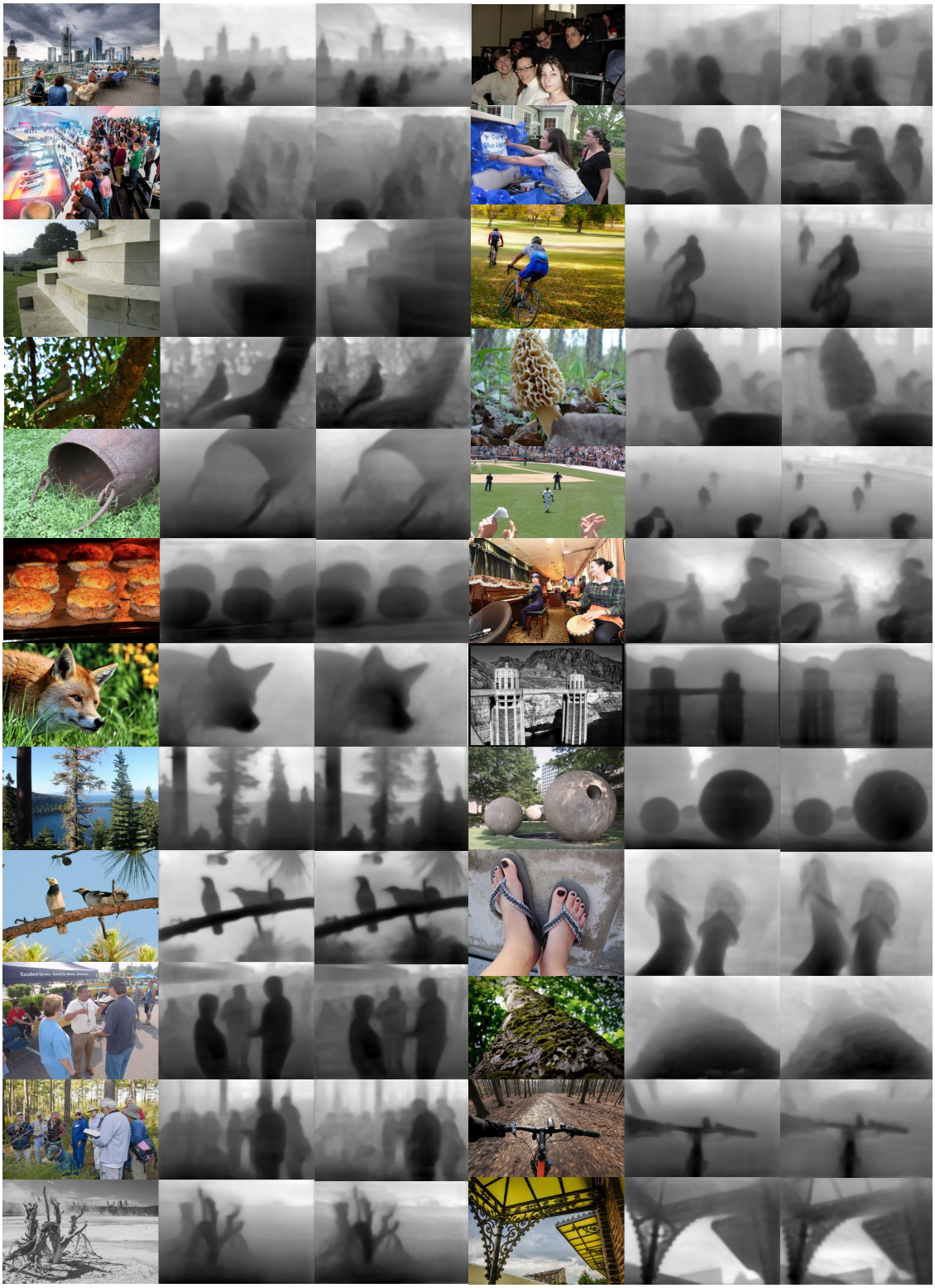



Figure 5.7: Qualitative results on the DIW test set by the EncDecResNet [17] trained on ImageNet + ReDWeb + DIW (*w/o YouTube3D*), and fine-tuned on YouTube3D (*w/ YouTube3D*).

# CHAPTER 6

# Conclusions and Future Work

## 6.1 Contributions

This dissertation has made contributions to single-view 3D perception in the wild in two major fronts. First, we have presented novel methods to collect large-scale 3D datasets from the Internet, effectively addressing the lack of diverse data issue in 3D vision (Chapter 2, 3, 4, 5). Second, based on the collected datasets we have proposed novel methods to advance single-view perception networks (Chapter 2, 3, 4).

### 6.1.1 3D Acquisition from the Internet

Data has played a major role in computer vision. Unlike the problem of image classification where large-scale datasets like ImageNet [27] have propelled significant progress, progress in single-view 3D perception has been largely hindered by the problem of lacking diverse and large-scale datasets. We have presented novel ways to acquire diverse 3D supervision from the Internet either through crowdsourcing or automated methods, and constructed four datasets: In Chapter 2, we propose a novel task of annotating relative depth, and construct Depth in the Wild. In Chapter 3, we propose a novel task of annotating surface normal on arbitrary surfaces, and construct Surface Normals in the Wild. In Chapter 4, we present a novel pipeline of creating dense 3D surfaces from human annotations, and construct Open Annotations of Single-Image Surfaces. In Chapter 5, we present a novel Quality Assessment Network to identify accurate SfM results on YouTube videos, and construct YouTube3D. These datasets cover vastly different aspects of 3D vision, ranging from depth, surface normals, to occlusion and fold, which are available for images in the wild either for the first time, or at a much large scale than prior work. They have served as valuable resources for benchmarking and training 3D perception algorithms in the wild.

### 6.1.2 Advancing Single-view 3D in the Wild

The availability of the aforementioned large-scale 3D datasets in the wild opens up a lot of research opportunities. Through benchmarking prior arts on them, we identified that even though methods trained on existing RGB-D datasets perform very well on prior benchmarks, they still perform poorly on arbitrary images in the wild across a suite of 3D perception tasks (Chapter 2, 3, 4). This finding points to the direction of leveraging 3D data acquired from the Internet to advance 3D perception. To that end, we propose novel training losses and problem formulations to train on the acquired data.

In Chapter 2, we have studied single-view depth estimation with relative depth from Depth in the Wild as supervision. We propose the novel task of learning to predict depth ordering instead of metric depth. This task is made possible by the introduction of a novel ranking loss. We show that by leveraging existing RGB-D data and the relative depth from Depth in the wild, we can significantly improve over the prior art of Eigen et al. [29].

In Chapter 3, we have studied surface normal estimations with a goal of using surface normal supervision to advance depth perception in the wild. Our novelty lies in proposing two losses, one that emphasizes depth accuracy, and another one that emphasizes surface normal accuracy. Our results suggest that with surface normals as supervision we can further advance depth perception in the wild.

In Chapter 4, we have studied a wide spectrum of single-view 3D tasks in the wild. We present a novel depth evaluation metric that takes into consideration the effect of focal lengths and perspective projection. In addition to that, through designing a novel annotation pipeline, we identify the importance of understanding occlusion and fold in 3D perception, and the benefits of knowing the relative normals. We thus present the novel tasks to jointly estimating occlusion and fold from a single image, and to estimate relative normals from a single image. By benchmarking state-of-the-art methods on these novel tasks, we find a significant gap between human and machine performance. Such finding points to new research opportunities.

## 6.2 Future Work

Despite recent progress, single-view 3D remains a challenging and unsolved problem. Based on the research presented in this dissertation, we identified some possible future research directions.

### 6.2.1 Aligning 3D Metrics with Human Perception

In Chapter 4, benchmarking prior works on OASIS has revealed limitations in some of the popular 3D metrics, as demonstrated by the fact that they do not align well with the human perception of 3D. As illustrated by Fig. 4.8, when large areas of uninteresting geometry are present, they tend to dominate the error metric. Any error in the small but perceptually important details is thus neglected.

Deigning metrics that pay special attention to the reconstruct details could help us better understand the limits of current single-view 3D methods, and develop better loss functions that are more aligned with human perception. The research questions would be to identify locations where humans deem important in recognizing and perceiving 3D. One possibility is to draw inspiration from the human annotation pipeline in Chapter 4, where occlusion, fold, and planarity has been critical in reconstructing 3D surfaces. For example, given a 3D reconstructions, an evaluation metric could be designed to reflect the following qualitative aspects: Are the occlusion boundaries between objects accurately recovered? Are the sharp changes in surface normals (i.e. fold), faithfully reflected in the reconstruction? Are regions that are supposed to be planar correctly reconstructed to be planar?

### 6.2.2 Automatic Mining of Dense 3D Supervision from the Internet

It remains labor-intensive and expensive to acquire high-quality 3D supervision with manual-labor regardless of how efficient the collection method is, as evident in Chapters 2, 3, and 4. Thus, automatically acquiring 3D is a more attractive alternative. Chapter 5 and other contemporary work [59] have explored using SfM to automatically mine 3D from Internet videos or image collections. However, their reconstructions are either sparse or are prone to error, with the main reason being that SfM is not robust against interference from moving objects and feature mismatches. One promising direction is to reconstruct small surfaces in the scene using SfM instead of trying to reconstruct a large part or the entirety of the scene. Ideally, 3D points on a small 3D surfaces could be assumed to be undergoing a uniform Euclidean transformation from frame to frame, even if they are on moving objects. Under this assumption, a SfM pipeline possesses the possibility to reliably recover correct dense reconstructions of small surfaces from videos. The research question would then involve how to filter out the bad reconstructions from the good ones in an automatic fashion.

### 6.2.3 Multi-stage Inferences of Single-view 3D

Most state-of-the-art single-view 3D estimation systems have largely been designed to be single-stage, meaning one image is fed into a neural network and a depth output is inferred end-to-end. However, an alternative way is to decompose the image-to-3D task into multiple stages,

where each stage is a much easier task than the original task, and we just need to solve them one by one. This idea has been explored in Chapter 4 with success, where an annotation pipeline is designed to allow humans to draw down various simple 3D cues one at a time, and from these cues, a complex 3D mesh is created. One promising direction is to design networks that embody this idea of multi-stage 3D inference, where a network learns to first decompose the image-to-3D task into multiple simpler ones and then from the solutions of these simpler tasks construct the 3D. The research question would then include identifying the stages and tasks for a neural network to learn so that better 3D can be inferred. A possible idea is to imitate the annotation pipeline in Chapter 4 with a neural network — train a network to recover occlusion, fold, surface normals, and relative depth, from which a final 3D mesh can be recovered.

### 6.2.4 Acquiring Completed 3D Reconstructions of Objects in the Wild

Single-view 3D data are usually in the form of depth maps, where ground truth 3D is available for the visible parts of objects, and that of the occluded part is missing. While 3D knowledge of visible parts is sufficient for most vision tasks, tasks such as planning and physics reasoning would get a significant boost if knowledge of the 3D in the occluded area were present. Therefore, machines should possess the ability to not only infer the 3D of visible parts, but also hallucinate 3D of occluded parts. Teaching machines to perform this task would require completed 3D reconstructions of objects in images as supervision. However, none of the existing datasets provide such data. One promising direction is to collect it from the Internet via crowdsourcing, based on the observation that the human mind is capable of performing this imagination task easily. The research question would be to design an efficient UI to allow workers to annotate the completed 3D reconstructions accurately and efficiently, as well as algorithms to infer completed 3D from a single image.

# BIBLIOGRAPHY


[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[3] M. H. Baig, V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Im2depth: Scalable exemplar based depth transfer. In *WACV*. IEEE, 2014.

[4] M. H. Baig and L. Torresani. Coupled depth learning. *arXiv preprint arXiv:1501.04537*, 2015.

[5] A. Bansal, A. Farhadi, and D. Parikh. Towards transparent systems: Semantic characterization of failure modes. In *European Conference on Computer Vision*, pages 366–381. Springer, 2014.

[6] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016.

[7] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *European Conference on Computer Vision*, pages 57–70. Springer, 2012.

[8] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015.

[9] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *TOG*, 2014.

[10] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Trans. on Graphics (SIGGRAPH)*, 32(4), 2013.

[11] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[13] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*. ACM, 2007.

[14] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *Advances in Neural Information Processing Systems*, pages 2658–2666, 2016.

[15] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[16] W. Chen and J. Deng. Learning single-image depth from videos using quality assessment networks. *arXiv preprint arXiv:1806.09573*, 2018.

[17] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in neural information processing systems*, pages 730–738, 2016.

[18] W. Chen, S. Qian, and J. Deng. Learning single-image depth from videos using quality assessment networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5604–5613, 2019.

[19] W. Chen, S. Qian, D. Fan, N. Kojima, M. Hamilton, and J. Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] W. Chen, D. Xiang, and J. Deng. Surface normals in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1557–1566, 2017.

[21] W. W.-C. Chiu, U. Blanke, and M. Fritz. Improving the kinect by cross-modal stereo. In *BMVC*, 2011.

[22] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016.

[23] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[25] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert. Introspective perception: Learning to predict failures in vision systems. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1743–1750. IEEE, 2016.

[26] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *arXiv preprint arXiv:1702.04405*, 2017.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[28] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[29] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[30] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017.

[31] S. Galliani and K. Schindler. Just look at the image: viewpoint-specific surface normal prediction for improved multi-view reconstruction. 2016.

[32] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[35] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv preprint arXiv:1609.03677*, 2016.

[36] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018.

[37] C. Hane, L. Ladicky, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *CVPR*, 2015.

[38] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[39] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. 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*, pages 3287–3295, 2015.

[40] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM transactions on graphics (TOG)*, volume 24, pages 577–584. ACM, 2005.

[41] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3):328–346, 2011.

[42] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. 1997.

[43] S. Ikehata, I. Boyadzhiev, Q. Shan, and Y. Furukawa. Panoramic structure from motion via geometric relationship detection. *arXiv preprint arXiv:1612.01256*, 2016.

[44] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*. Springer, 2013.

[45] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[46] K. Karsch, Z. Liao, J. Rock, J. T. Barron, and D. Hoiem. Boundary cues for 3d object shape recovery. In *CVPR*, 2013.

[47] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[48] K. Karsch, C. Liu, and S. B. Kang. Depthtransfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 2014.

[49] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. SGP. Eurographics Association, 2006.

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

[51] J. J. Koenderink, A. J. Van Doorn, and A. M. Kappers. Surface perception in pictures. *Attention, Perception, & Psychophysics*, 52(5):487–496, 1992.

[52] P. Krähenbühl. Free supervision from video games. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2955–2964, 2018.

[53] A. Kushal and S. M. Seitz. Single view reconstruction of piecewise swept surfaces. In *2013 International Conference on 3D Vision-3DV 2013*, pages 239–246. IEEE, 2013.

[54] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*. IEEE, 2014.

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

[56] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[57] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143. IEEE, 2009.

[58] 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 *CVPR*, 2015.

[59] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[60] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Computer Graphics Forum*, volume 18, pages 39–50. Wiley Online Library, 1999.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[62] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260. IEEE, 2010.

[63] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. Planercnn: 3d plane detection and reconstruction from a single image. *arXiv preprint arXiv:1812.04072*, 2018.

[64] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018.

[65] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[66] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[67] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation.

In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[68] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. 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*, pages 4040–4048, 2016.

[69] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv preprint arXiv:1612.05079*, 2016.

[70] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[71] T. Narihira, M. Maire, and S. X. Yu. Learning lightness from human judgement on relative reflectance. In *CVPR*. IEEE, 2015.

[72] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *arXiv preprint arXiv:1603.06937*, 2016.

[73] D. Parikh and K. Grauman. Relative attributes. In *ICCV*. IEEE, 2011.

[74] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[75] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[76] Y. Quéau, J.-D. Durou, and J.-F. Aujol. Normal integration: a survey. *Journal of Mathematical Imaging and Vision*, 60(4):576–593, 2018.

[77] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz. Soccer on your tabletop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4738–4747, 2018.

[78] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[80] B. C. Russell and A. Torralba. Building a database of 3d scenes from user annotations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2711–2718. IEEE, 2009.

[81] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[82] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[83] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International journal of computer vision*, 76(1):53–69, 2008.

[84] A. Saxena, M. Sun, and A. Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 2009.

[85] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[86] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2016.

[87] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proc. CVPR*, volume 3, 2017.

[88] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[89] E. Shelhamer, J. Barron, and T. Darrell. Scene intrinsics and depth from a single image. In *ICCV Workshops*, 2015.

[90] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *TOG*, 2015.

[91] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*. Springer, 2012.

[92] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[93] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

[94] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *arXiv preprint arXiv:1611.08974*, 2016.

[95] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[96] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *International journal of computer vision*, 82(3):325, 2009.

[97] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018.

[98] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[99] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[100] J. T. Todd and J. F. Norman. The visual perception of 3-d shape from multiple cues: Are observers capable of perceiving metric structure? *Perception & Psychophysics*, pages 31–47, 2003.

[101] S. Tulsiani, S. Gupta, D. Fouhey, A. A. Efros, and J. Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. *arXiv preprint arXiv:1712.01812*, 2017.

[102] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[103] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[104] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[105] P. Wang, X. Shen, B. Russell, S. Cohen, B. Price, and A. L. Yuille. Surge: Surface regularized geometry estimation from a single image. In *Advances in Neural Information Processing Systems*, pages 172–180, 2016.

[106] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[107] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[108] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018.

[109] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3d: A large scale database for 3d object recognition. In *European Conference on Computer Vision*, pages 160–176. Springer, 2016.

[110] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016.

[111] S. Xie and Z. Tu. Holistically-nested edge detection. *CoRR*, abs/1504.06375, 2015.

[112] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[113] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler. From shading to local shape. *TPAMI*, 2015.

[114] X. Xu, D. Sun, S. Liu, W. Ren, Y.-J. Zhang, M.-H. Yang, and J. Sun. Rendering portraitures from monocular camera and beyond. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–50, 2018.

[115] F. Yang and Z. Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[116] Z. Yu, J. Zheng, D. Lian, Z. Zhou, and S. Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. *arXiv preprint arXiv:1902.09777*, 2019.

[117] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3566–3573, 2014.

[118] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5057–5065. IEEE, 2017.

[119] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[120] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015.

[121] T. Zhou, P. Krahenbuhl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*, 2015.

[122] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015.

[123] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *ICCV*, 2015.