3D Surfaces in the Wild

David Fan

Advisor: Professor Jia Deng

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Computer Science

Princeton University

May 2019

Abstract

The recovery of 3D structure from a single 2D image remains an open problem in computer vision. Neural networks do reasonably well at predicting the 3D structure of limited scenes - mostly of indoor scenes and road scenes. But, they are unable to generalize well to unseen training images. We hypothesize that this is in large part due to the lack of diverse and large scale training data for 3D inference. Recent work has attempted to crowdsource 3D annotations of images in the "wild", but due to the large amount of labor involved, fails to produce datasets that are large and expressive enough to improve state-of-art in 3D inference.

Our contribution is three-fold. First, we present a methodology for efficiently obtaining dense 3D annotations of everyday images scraped from the Internet, or images in the wild. Applying this method to Amazon Mechanical Turk workers, we crowdsourced a novel 3D vision dataset of large scale and diversity, which we call "3SIW". We provide full surface normal, depth, fold boundary, and occlusion boundary annotations for 20,000 images from the wild. Our methodology can be used to create other datasets of larger scale and diversity. Secondly, we provide benchmarks on 3SIW for four tasks: surface normal estimation, occlusion detection, fold detection, and semantic segmentation of planar surfaces. Lastly, we demonstrate that training on larger and more diverse data advances the state-of-art in 3D visual systems.

Acknowledgments

Around a year ago, I was debating whether or not to do a senior thesis. I am so glad that I decided to do it, and would like to properly thank everyone for making this experience possible.

First, I would like to thank Professor Jia Deng for allowing me to work with the Princeton Vision and Learning lab. I first became interested in deep learning and computer vision during the middle of junior spring, but my research experiences up until that point had been in physics and computational biology. Entering senior year, I did not have a lot of practical experience with deep learning or computer vision. Professor Deng nevertheless saw potential in me and took a chance, for which I am extremely grateful.

Working on this project over the past nine months has been extremely rewarding; I've had the chance to design and implement data pipelines end-to-end, solve complex engineering challenges, and train state-of-art models in novel settings, among other things. Not a single day has passed without immense learning and challenges, and I would like to thank all of my research collaborators for their support and guidance. I worked most closely day-to-day with Weifeng Chen, without whom this thesis could not have happened. I am extremely greatful for his guidance, patience, criticism, and interesting discussions. I would like to thank Shengyi Qian, Noriyuki Kojima, and Max Hamilton of the University of Michigan, for continually impressing me with their domain expertise and ability to get things done. I would also like to thank Hei Law for always being on top of Ionic cluster outages, and providing the lab with up-to-date information about computing resources. To the rest of PVL: thank you for answering my questions, as well as leading challenging conversations during weekly lab meetings. I have learned a lot from you.

Doing research at Princeton can get lonely at times, but fortunately I have enjoyed the company of some amazing friends and colleagues. To Yun Teng and Matthew Li,

who I affectionately name the “Deng Bros”: thank you for your camaraderie, support, and friendship. Research is infinitely more fun in your company, and I am so proud of what you have managed to accomplish this year. In the same capacity, I would like to thank Yannis Karakozis and Berthy Feng for stimulating discussions, as well as continually inspiring me to improve as a researcher. Finally, I would like to thank Professor Olga Russakovsky for being my second reader, giving amazing lectures in graduate computer vision this spring, and inspiring me with her leadership in making AI accessible to minorities.

Entering Princeton, I never imagined that I would found my own campus group, or direct a biannual hackathon hosting over 1,000 people. Thank you to the members of Science Olympiad, HackPrinceton, and the rest of E-Club for gracing me with your passion, intellect, resourcefullness, and kindness. Magical things happen when people with purpose and vision put their heads together towards a common cause, and I’m glad that we got a chance to work together. Thank you to the members of ping pong club, and in particular Theodore Ando, for motivating me to occasionally go to the gym. Many other people have made my Princeton experience special, and it would be impossible to list everyone. I’ve learned something from everyone I’ve met, and am grateful to my friends and roommates for their continual support and love. In particular, I’d like to thank Roland Fong, Mayee Chen, Lillian Xu, Jessica Ho, Casey Chow, Peter Chen, Yang Song, Frances Ling, and Alex Xu for always having my back. To Nikil Pancha, Lucy Zhang, and Junlan Lu: thank you for keeping in touch over the years, even when I was buried under work and unresponsive. To my two thesis fairies Kat Song and Lily Zhang: thank you for showering me with tasy snacks and drinks this semester that I didn’t deserve – I really appreciate your generosity. Many thanks go to Evan Chow, Nathan Wei, Jonathan Lu, Vincent Po, Nick Chow, and Jisung Kim for their mentorship over the past years.

My passion for science began early on, and matured during high school. I am indebted to Ms. Hallie Kleinfield, Mr. Jason Sullivan, and Mr. Chris Resch for my time with the Montgomery Science Olympiad team, which instilled in me the importance of coupling intelligence with an even greater work ethic, and holding high aspirations. I also extend a heartfelt thank you to my past research mentors who have blessed me with amazing opportunities that broadened my perspectives: Professors Sang-Wook Cheong, Yongkyu Park, Isaac Kohane, Raj Manrai, and Dr. Jean Fan.

Lastly, I have to thank my parents and sister for raising me and teaching me all the good things that I know. I would not have gotten here today without their love and support. My success is all their's.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Humans see everything through 2D projections on the retina, and yet are effortlessly able to perceive the world in 3D. Without much thought, we can understand, interact with, and reason about our surroundings. For instance, we know that a tennis ball is round, even if we have neither seen nor touched one before. When walking through a city, we recognize that buildings in the horizon are further from us than buildings on the next block. We also know that all buildings are attached to the ground, even if we can only see the top of buildings. Humans are able to flawlessly perceive the world in 3D from 2D visual inputs.

Unfortunately, replicating this visual prowess in computer systems is difficult. Single-view 3D reconstruction, or the recovery of 3D structure from a single 2D image, remains an open problem in computer vision and is the topic of this thesis. The problem is inherently ambiguous; an infinite number of possible 3D geometries correspond to any given 2D image (Figure 1.1). Despite this, humans are able to infer the most probable (and correct) representation using prior knowledge of the world, but computers have no such priors. It stands to reason that approaches which give computers these priors, or allow computers to learn priors a posteriori, can help computers perceive the 3D world in a similar capacity to humans.

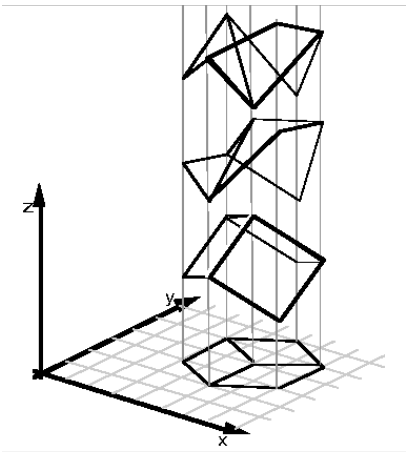


Figure 1.1: An 2D image corresponds to infinitely many possible 3D structures. Borrowed from [48].

Early work on single-view 3D reconstruction uses hand-designed priors to make assumptions about the scene. [38] encoded these priors in terms of specific constraints on the point-to-point correspondence between the 3D structure and 2D projected image. [24, 2] made statistical assumptions about the image's structure. Such methods do not model real surfaces well. Recent improvements in hardware and computing power have enabled an alternative data-driven approach, which leverages the adaptability of deep neural networks to learn priors, by training on images with known ground truth 3D structures.

The success of deep learning on any task is highly dependent on the quality and quantity of training data available; neural networks are only as good as the data that they learned from. Convolutional neural networks (CNNs) have recently outperformed previous state-of-art machine learning techniques on a number of tasks, including image classification [22, 47, 30, 12], object detection [19, 21, 45, 31], semantic image segmentation [8, 55], and human pose estimation [42, 53].

While neural networks are also state-of-art on 3D vision tasks, including single-image depth estimation [14, 9] and surface normal estimation [14, 10], they have not yet achieved their full potential. It is difficult to obtain 3D supervision at scale; exist-

ing datasets for 3D vision are either small or do not include a diverse range of images. Thus, visual systems trained on these datasets perform poorly on images outside of the training set. In other words, they have difficulty generalizing to everyday images, or images from the "wild". For example, consider Figure 1.2. The network by Eigen et al. [14] is state-of-art on the NYU Depth V2 dataset [46] for single-image depth estimation, but struggles on outdoor images, because it was trained only on indoor images. Generalization gaps such as these can be improved upon by creating larger-scale datasets with greater heterogeneity for 3D vision.

a) NYU Image 1 b) NYU Image 2 c) Internet Image 1 d) Internet Image 2

Figure 1.2: A network trained on the NYU Depth dataset [46] predicts depth for two indoor images, and two outdoor images. Darker pixel intensity denotes that the object is closer. Note that the network mistakenly predicts the dog's head in c), and the building's door in d) to be further away than other parts of the image. This may be because many images in the NYU dataset have white walls, which are the furthest objects in those scenes. Networks trained on datasets with limited diversity fail to learn features that are fully representative of the real world.

What properties should a comprehensive dataset for 3D vision contain? The 3D shape of any given object is determined by: depth, surface normal orientation, occlusion boundaries (depth discontinuities), and fold boundaries (surface normal discontinuities). Existing datasets provide two or fewer of these properties, or a limited selection of images. A popular dataset for depth and surface normal estimation is the aforementioned NYU V2 dataset [46], but it exclusively contains images of indoor

scenes. This is because the dataset was collected using depth sensors, which are most accurate in a controlled indoor environment. A newer dataset for depth estimation is the Youtube3D dataset [8], which leverages the heterogeneity of online videos, but only provides depth ground truths. A popular dataset for occlusion estimation is the Carnegie Mellon video dataset [50], but it only has 30 scenes. No dataset currently exists which contains all relevant 3D properties, and also fully represents the diversity of wild images at scale. As a result, current state-of-art networks trained on these limited datasets have a difficult time generalizing to unseen scenes.

Recent work by Chen et al. [9, 10] is the first to address this issue by crowdsourcing the collection of ground truth 3D annotations from Internet images. They crowdsourced two datasets, the first of which is "Depth in the Wild"(DIW) [9], which provides relative-depth ground truths — for each image, two pixels and their ordinal relationship (i.e. is point x closer or further than y from the viewer) are known. Their second dataset is "Surface Normals in the Wild"(SNOW) [10], which provides the surface normal orientation of one pixel per image. These two datasets are the only 3D vision datasets with images in the wild currently in existence. Both DIW and SNOW are large and highly diverse because they contain images from the wild, and are not limited to a particular type of scene. But, both are also "sparse", because they only provide ground truth annotations for one or two pixels per image.

Chen et al. demonstrate that training on these sparse but more diverse datasets leads to nearly equivalent results, compared to training on datasets with dense pixel-wise ground truths but less representative images. While this is impressive, sparse datasets only give weak training signals. Supervised learning does strictly better when given pixel-wise ground truths, because the training signal is stronger and less noisy. However, collecting dense annotations can be difficult, expensive, and labor-intensive. The goal of this thesis is to 1) collect a densely annotated 3D vision dataset at scale, and 2) demonstrate its usefulness for a variety of 3D vision tasks.

We first created an annotation pipeline which efficiently provides dense 3D annotations at scale for images in the wild. Section 3 addresses our approach in detail and walks through the annotation UI. This annotation process yields the following ground truths which fully define the 3D structure of an object: full surface normal and depth maps, occlusion boundaries, fold boundaries, instance planar segmentations (which pixels belong to which plane), semantic surface segmentations (which pixels belong to planar vs. curved surfaces), and planarity relationships (given two planes x and y, are x and y parallel, perpendicular, or neither). Given these human-annotated ground truths, we then explore the question of how well neural networks can learn to approximate 3D reconstructions in Section 5.

To summarize, our contributions to the literature are three-fold. First, we present a novel methodology for efficiently crowd-sourcing dense 3D annotations of images in the wild, featuring a diverse and novel array of 3D properties. Secondly, we provide performance benchmarks for four of the 3D vision tasks enabled by our dataset: surface normal estimation, occlusion boundary detection, fold boundary detection, and semantic segmentation of planar surfaces. The latter two tasks are novel, and we attain new state-of-results on the first task, demonstrating the utility of collecting and training on better data. This constitutes the first-ever large-scale benchmark for single-view 3D vision tasks on heterogenous images from the wild.

Chapter 2

Background and Related Work

2.1 Cues for Recovering 3D Shape

A variety of image properties ("cues") inform 3D reconstruction. Earlier work focused on using shading cues, or lighting differences, to infer surface normal orientation and depth. The intuition is that the amount of light reflected by a surface depends on the surface orientation, in relation to the viewer and light source. However, shape-from-shading often fails to recover meaningful structural information. Under certain lighting conditions, even the convexity or concavity of a surface recovered from shape-from-shading is ambiguous [43]. That is, lighting patterns do not independently define the shape of an object, and depend in part on the environment. This is problematic as a variety of lighting conditions and viewing angles exist in the real world, and not all are ideal. A data collection method that depends on shape-from-shading is not robust and not scalable.

In contrast, fold boundaries and occlusion boundaries (which define surface normal discontinuity and depth discontinuity, respectively) provide useful constraints on object geometry that hold independently of viewing direction, lighting conditions, and other environmental factors. For any image, knowing the surface normal orien-

tation and depth per pixel provides a rough 2.5D representation, in that the shape is 3-dimensional, but the entire topology is connected. There are no overlapping nor overhanging elements, but real life scenes are composed of both separated and connected regions. Humans do not perceive the 3D world as point-wise depth maps, but rather as separate regions whose spatial and geometric relationships are delineated by occlusions and folds. Occlusions and folds tell us about the relationship between surfaces, and whether surfaces are connected or not. Thus, folds and occlusions are essential for attaining human-level 3D perception.

Karsch et al. [27] validate this intuition by demonstrating that 3D shape reconstruction improves when incorporating folds and occlusions into a shape-from-shading approach. Taking inspiration from them, we also collect occlusion and fold boundary annotations that segment the image into distinct regions. However, instead of inferring depth and surface normals through shading cues, which are unreliable for aforementioned reasons, we directly collect sparse surface normals. We then recover full pixel-wise depth and surface normals from the sparse surface normal annotations. Section 3 discusses our annotation process and shape recovery in detail. Our annotation process yields full ground truths that define the 3D geometry of an object, and serve as useful training data.

2.2 Approaches to Collecting RGB-3D Datasets

RGB-3D datasets are datasets containing ground truth 3D properties of RGB images. There are four approaches for assembling such datasets in the literature. One such approach involves using depth sensors [11, 46, 18]. However, depth sensors have limitations. Microsoft Kinect [54] is a popular sensor used to assemble datasets such as ScanNet [11] and NYU Depth [46], and measures depth by triangulating an infrared laser's diffraction pattern against a reference pattern. This method is fairly accurate

for small distances, but the error increases rapidly for longer distances. Measurements are also sensitive to strong lighting conditions, which can result in gaps with no measured depth [29]. Due to these limitations, depth datasets collected through sensors are currently limited to indoor [11, 46, 18] and driving scenes [18].

A second approach is to synthesize 3D structures using computer graphics [5, 39, 40, 49]. However, it is difficult to synthesize enough shapes to fully represent the diversity of shapes in nature. Many shapes are slight variations on or combinations of other shapes, and enumerating all possibilities is intractable. Furthermore, synthetic scene rendering currently requires artistic expertise (e.g. to determine shading and object reflectance), and a limited number of 3D assets prevents fully realistic scene rendering.

A third approach is to use multi-view stereo and modern structure from motion (SfM) methods to recover depth from multiple images of the same scene [8, 34]. While this approach works well for static scenes, it fails on dynamic scenes in which an object is moving (such as a person running, or a car driving down the street). An ideal method for dataset collection should work for both static and non-static scenes. In addition, one flaw of this approach is that it requires multiple views of each scene. In practice, there are rarely multiple photos available of the same scene. Collecting multiple images of a scene takes additional time, storage space, and money. Thus, an ideal dataset collection method operates off single images.

The fourth and last approach is to crowdsource ground truth annotations, as mentioned in the introduction. Crowdsourcing is an appealing approach because it intrinsically enables scalability. The above three approaches are not easily scalable for the aforementioned reasons. In addition, crowdsourcing leverages the human visual system and its ability to perceive shape, which is arguably more reliable than any algorithm. A goal of computer vision after all, is to replicate human visual performance in computer systems. By using humans to collect ground truths, we

directly teach computers to learn priors that humans use. One drawback of this approach is that crowdsourced data may vary drastically in accuracy and quality; people online may not follow instructions consistently and faithfully. Receiving bad data often would negate any benefits in scalability gained by crowdsourcing, because additional time then needs to be spent removing that data. However, the success of other recent works in crowdsourcing 3D annotations [3, 9, 10] gave us reason to believe in this approach.

2.3 Summary of Existing Datasets for 3D Vision

Existing datasets for 3D vision are limited in that they either lack scale, only provide one or two types of ground truth annotations, or contain only a particular type of scene (i.e. indoor or road driving). Table 2.1 summarizes the most popular datasets for 3D vision, and their respective limitations and strengths. Most datasets are for depth and surface normals, and only one commonly used dataset exists for occlusions. Our dataset expands the range of 3D vision tasks possible, by offering a novel set of 3D annotations previously unavailable.

Dataset	Size	3D Annotations	Collection Method	Scene Diversity
NYU V2 Depth	407,024 frames / 464 scenes	Dense Depth, Surface Normals	Microsoft Kinect	Indoor Room
KITTI	93 frames	Dense Depth	Depth Sensor	Outdoor Driving
Make3D	534 images	Dense Depth	Depth Sensor, Synthetic, Multi-view Stereo	**Wild**
DIW	495,000 images	Sparse Depth	Crowdsourcing	**Wild**
SNOW	60,061 images	Sparse Surface Normals	Crowdsourcing	**Wild**
CMU Occlusion	30 scenes	Occlusions	Crowdsourcing	**Wild**
3SIW (Ours)	20,000+ images	Dense Depth, Surface Normals, Folds, Occlusions, Semantic and Instance Segmentations, Planarity Relationships	Crowdsourcing	**Wild**

Table 2.1: Summary of existing datasets for 3D vision and their respective limitations and strengths. Some offer dense annotations at scale, but for limited scenes. Some offer sparse annotations at scale, but for a diverse range of scenes. Some do not achieve scale at all. Our dataset addresses these limitations, and offers dense annotations at scale for diverse scenes from the wild.

2.4 State-of-Art in Single-View 3D Reconstruction

There are three existing tasks in the literature for single-view 3D reconstruction: 1) depth estimation, 2) surface normal estimation, and 3) occlusion boundary estimation. Convolutional neural networks are state-of-art in all three, and recent work has significantly improved accuracy on these tasks individually [32, 37, 51, 14, 33, 28, 25, 56, 17]. These methods consider each task independently; given an image, the network either directly predicts depth or surface normals. A new idea in the literature has been to jointly predict depth and surface normals, taking into account their intrinsic geometric relationships. Surface normals can be recovered from depth via a least square solution, and depth can be inferred from surface normals via a linear system of equations. Qi et al. [44] take this approach and first predict depth using one network, and then surface normals using another network. Then, they jointly optimize depth and surface normal using each other as the input. This work demonstrates that supervising on multiple 3D properties, rather than just one, can significantly advance the state-of-art in single-view 3D reconstruction. It also demonstrates that improvements in the estimation of one property, directly improve the estimation of other related 3D properties. This intuition motivates the creation of our dataset, which provides dense multi-modal 3D annotations at scale for images in the wild.

Chapter 3

Scalable Dense 3D Annotation Collection

This chapter presents the entire ground truth collection pipeline, from the Amazon Mechanical Turk worker qualification exam, to the annotation user interface, quality control and compensation mechanisms, and finally, the optimization problem that recovers full ground-truth metric depth and surface normals from sparse annotations. We begin by discussing general challenges behind data collection, to provide intuition for our implementation.

3.1 Challenges and Objectives

Crowdsourcing is the process of giving a single task for many random people to do, typically over the Internet. A general challenge of crowdsourcing is ensuring that these random individuals:

1. Understand what you want them to do.
2. Do the task correctly.
3. Are motivated financially to repeatedly do the task.

The first two objectives are obviously important, because incorrect datasets provide no value. The third objective is important because slow data collection hinders scalability. An ideal collection scheme is fast yet yields few errors; every bad worker submission wastes time and money, because it then has to be removed from the dataset. However, without spending enough time and money to ensure that submissions are of high quality, the first two objectives cannot be achieved. Thus, developing a good crowdsourcing pipeline requires striking a balance between speed (i.e how long it takes a worker to do the task), and quality (submissions are generally higher quality if workers spend more time per submission). Speed and cost are inversely proportional because workers expect to be compensated more for harder and longer tasks. Money and time are obviously finite in practice, so minimizing both is desirable.

In Section 3.3, we present a qualification quiz that addresses objective 1. In Section 3.5, we present a peer-driven quality control mechanism that addresses objective 2. Finally, in Section 3.6, we discuss a variety of compensation schemes, and the one that worked best for us.

3.2 Amazon Mechanical Turk

Amazon Mechanical Turk [4] is an online marketplace for human labor that has recently become popular in the research community, for its relatively low cost and high throughput. Numerous influential computer vision datasets have been collected with the help of Amazon Mechanical Turk, including ImageNet, Microsoft COCO, and Pascal VOC [12, 36, 15].

Workers who have registered with Mechanical Turk see a listing of all publically available HITs ("Human Intelligence Tasks"), and can choose which ones to do based on the description and award. We created two HITs: one for annotating images (providing 3D ground truths), and another for checking the correctness of submissions

to the prior task. Both are prefaced with a quiz, which workers must pass in order to do the HIT and receive compensation.

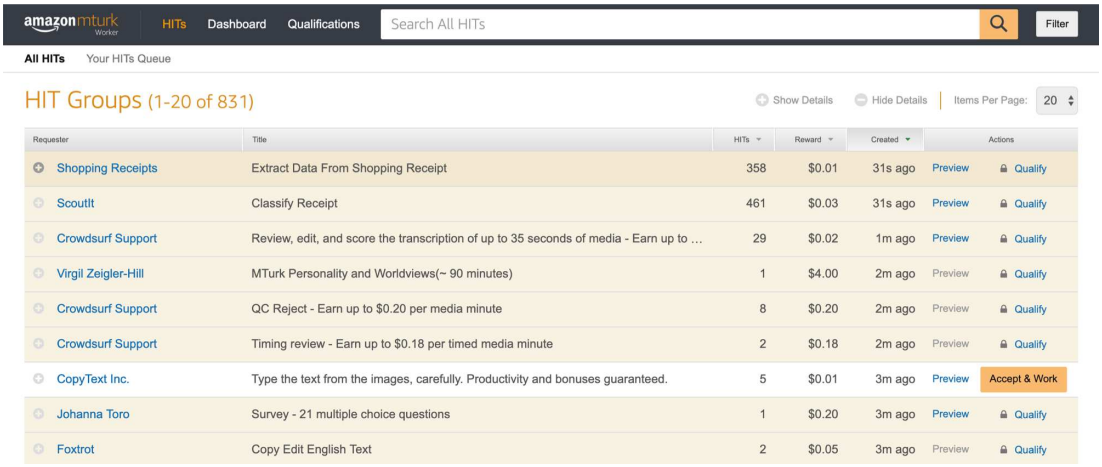


Figure 3.1: The Mechanical Turk homepage and a selection of available HITs.

3.3 Annotation HIT: Worker Qualification

The very first iteration of our annotation HIT did not include a qualification quiz because we wanted to see if workers would be diligent enough to read the tutorial page. Data collection is strictly faster without a qualification quiz, and we had hoped that the quality of submissions would be high enough without a quiz. However, it turned out that workers either didn't read the tutorial, or did not understand it completely, since most of the initial annotations were incorrect. Many of the 3D structures were completely flat and ignored geometric details. On the other extreme, a few workers understood the instructions, but paid too much attention to detail. For example, instead of treating a lawn as a flat surface, they attempted to annotate every individual blade of grass. Submitting too many details crashes our backend program. Thus, we realized that a mandatory qualification quiz was necessary both to ensure that workers understand the HIT, and also have the same standards with regards to how much detail is necessary.

Below is a screenshot of what a worker sees when they accept our annotation HIT for the first time. They must receive a threshold score of 85% in order to pass the quiz, and pass within three attempts. Workers who fail three times in a row are automatically blocked from visiting our HITs in the future. Workers who pass the quiz can directly do the annotation HIT in subsequent visits. Our quiz walks workers through the entire process of using our UI to annotate an image, and asks intermediate questions along the way to test their understanding.

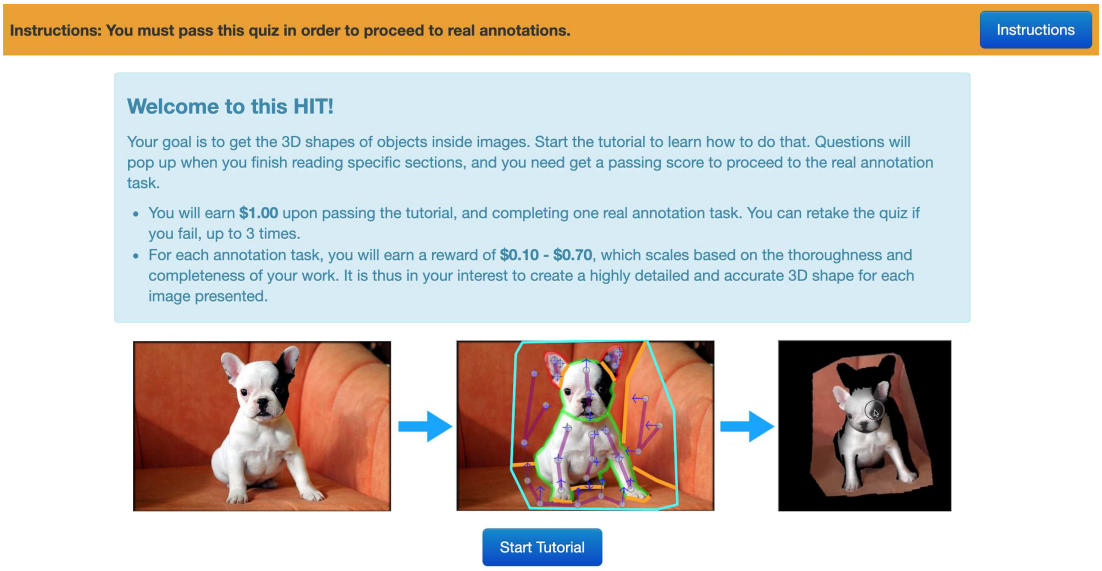
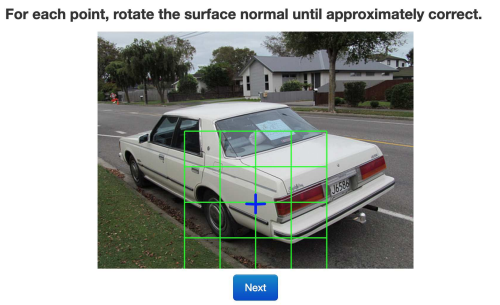


Figure 3.2: Worker qualification quiz for annotation HIT.

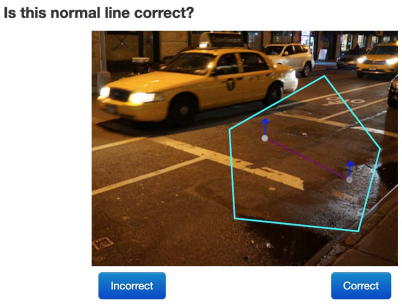
We were initially concerned that workers would be unwilling to take the quiz and opt for easier HITs instead. After trying a variety of price points, we found that awarding $1 for passing the quiz was the minimum needed to encourage workers to do the quiz correctly, and thus qualify for our annotation HIT. Coincidentally, [20] reports that the median income per hour on Amazon Mechanical Turk is approximately $2. If a worker passes our quiz in fifteen minutes, then they earn approximately twice as much money as the average worker, in that time. In practice, we found that of workers who passed the quiz within three attempts, most required no more than 20 minutes to do so.

Below in Figure 3.3 is a sample of the questions. Our quiz was created in response to a few common points of error. Workers often confuse fold boundaries and occlusions, however the distinction is key. Fold boundaries define discontinuities in surface normals (i.e. the sharpness of an edge), while occlusion boundaries define discontinuities in depth (i.e. separation between regions). Because occlusion boundaries define depth discontinuity, one side of the occlusion boundary must necessarily be closer to the viewer than the other side. We ask that workers draw occlusion lines in the direction such that the closer side lies on the right. As mentioned earlier, some workers annotate too many details; we also provide a question to calibrate standards on what is reasonable.

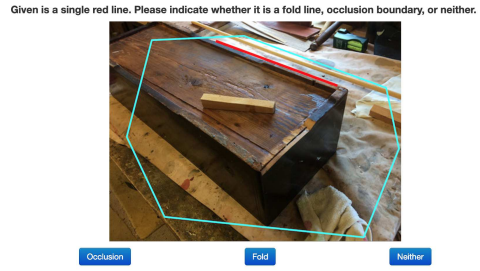
Only 8.5% of workers who attempted the quiz failed three times, which means that our passing score threshold was reasonable. The quiz is sufficiently hard as it fully walks through the annotation process, and asks questions at each step that address all potential failure points. Next, we present the annotation user interface (UI).



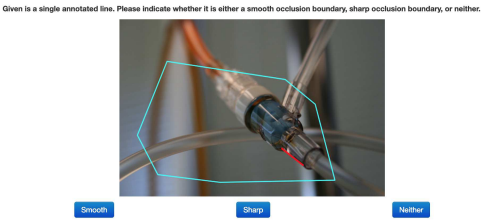
(a) Users must rotate the surface normal to be perpendicular to the surface.



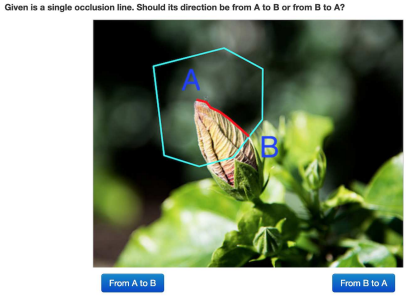
(b) Surface normals lines must be on the same connected surface.



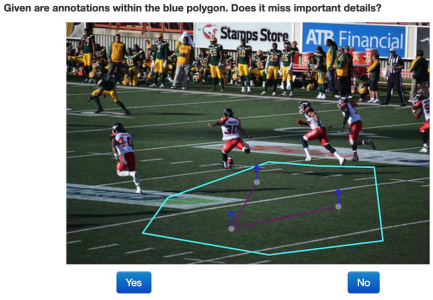
(c) There are two general types of boundaries: folds and occlusions.



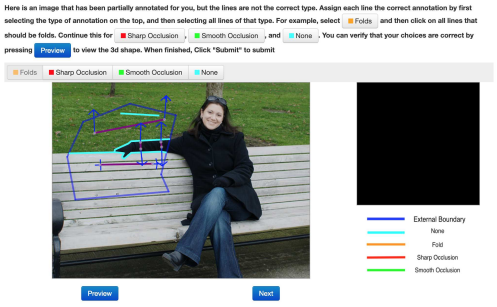
(d) Occlusions can be further divided into two categories: smooth and sharp.



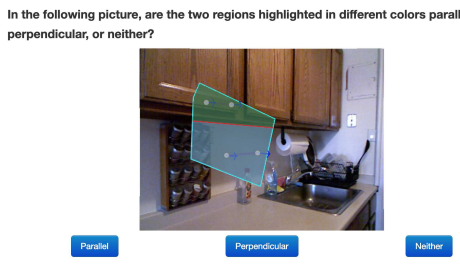
(e) Occlusions must be drawn in a consistent direction, in order to denote which surface is closer to the viewer.



(f) Users sometimes annotate too many details, such as individual blades of grass or water droplets.



(g) An incorrect annotation is given, and the user must correctly indicate which lines are folds, not occlusions.



(h) Users are asked whether pairs of planar surfaces are perpendicular, parallel, or neither in the UI.

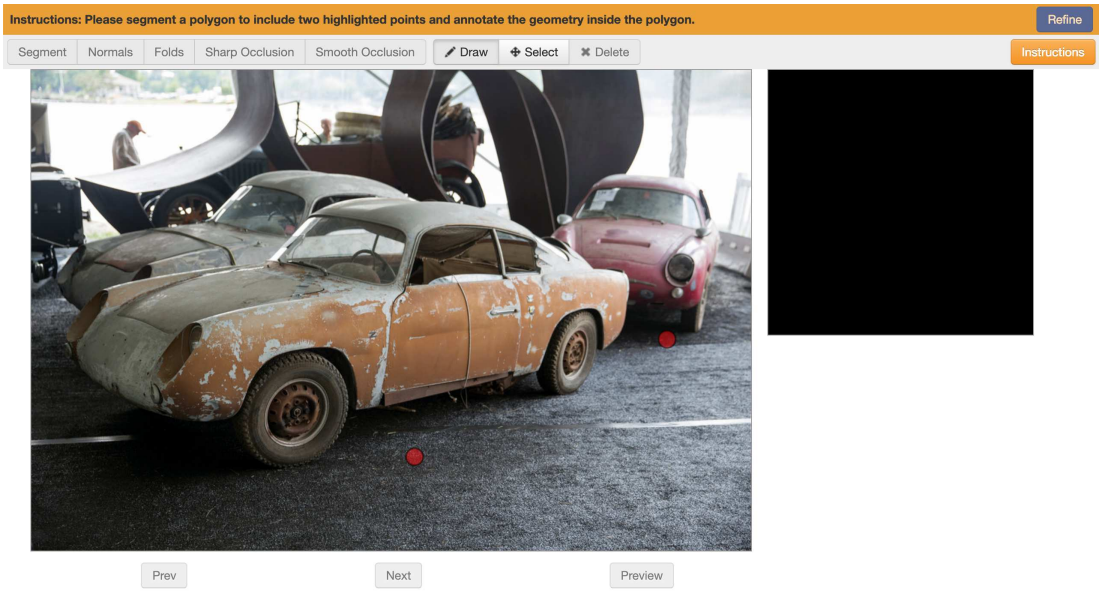
Figure 3.3: Sample annotation HIT qualification quiz questions.

3.4 Annotation HIT: UI

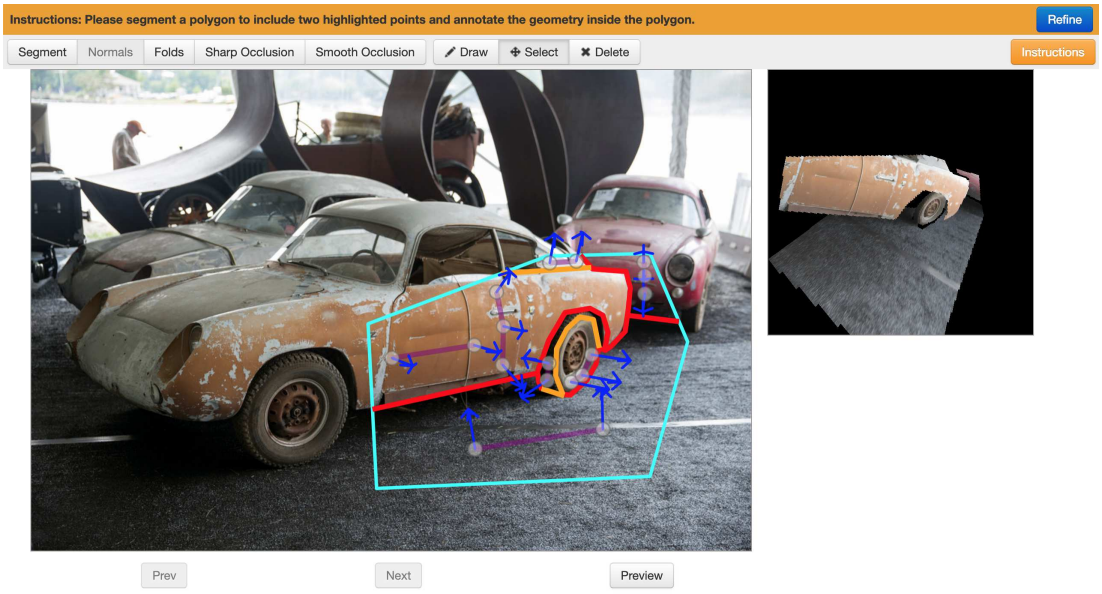
Most of the work in this section was done by Shengyi Qian (a student at the University of Michigan). His UI is presented in order to contextualize how the data collection process works, after a worker passes the quiz.

When a worker visits the annotation HIT (after passing the quiz), they are presented with one of 100,000 random images scraped from Flickr under a Creative Commons license. An image is not presented if it has already been annotated. Figure 3.4a shows the initial state of the UI, and Figure 3.4b shows the completed UI after a sample annotation.

The first step is to draw a polygon, denoting the region to be annotated. Next, the user segments the image by drawing fold boundaries, and smooth and sharp occlusion boundaries. Next, the user draws surface normals in each segmented region, to approximate its curvature. A region can either be planar (i.e. a flat surface), or curved. The UI will also ask the user about the planarity relationship between each plane (parallel, perpendicular, neither). When the user is satisfied, he/she can preview the resultant 3D structures and make adjustments until the 3D shape looks correct. Figure 3.5 shows every step.

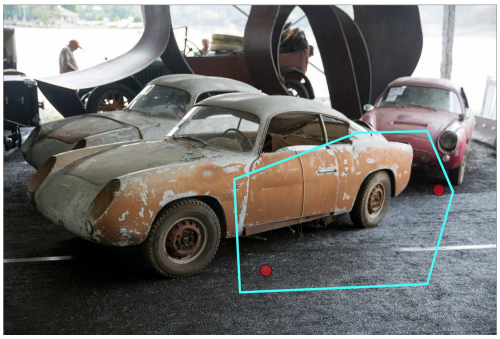


(a) UI when a user first loads the page.

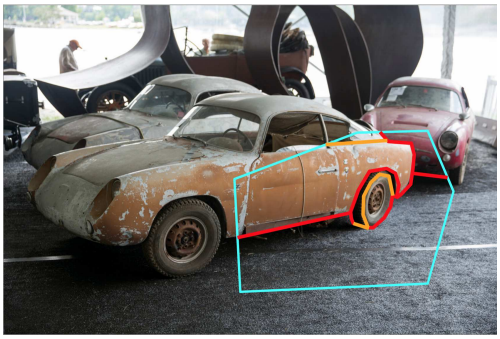


(b) Annotation UI when a user has fully annotated an image.

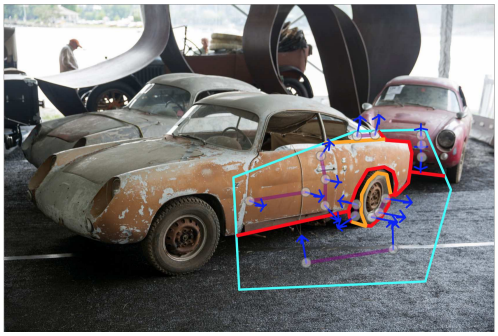
Figure 3.4: Annotation UI appearence.



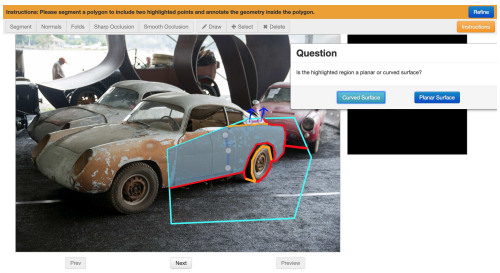
(a) The user first draws a boundary defining the region to be annotated.



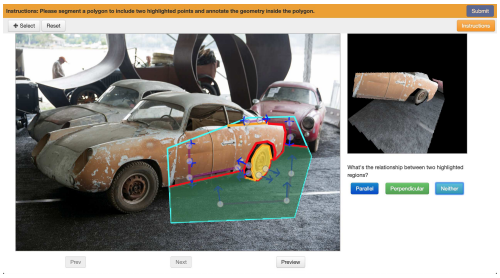
(b) Next, the user segments the image with folds and occlusions.



(c) Next, the user draws surface normals for each region.



(d) The user will be asked if each region is planar or curved.



(e) The user will be asked if every pair of regions is perpendicular or parallel.

(f) The final computed 3D structure using user's inputs.

Figure 3.5: Steps of annotating an image.

The preview button is important because it allows a worker to visualize the 3D structure they have annotated, and adjust their submission until the 3D structure makes sense. When a worker is satisfied, they submit and receive a new random image.

3.5 Evaluation HIT: UI

As mentioned in Section 3.1, the first two objectives were ensuring that workers understand the task, and can perform it correctly. Without a mechanism to review quality, some ground truth 3D structures will be incorrect and thus provide no value for supervised training. We implemented a peer-review system in which a user is presented with an image-3D structure pair, and asked to assign a label to each submission. There are four possible labels: "perfect", "very good", "acceptable", and "poor". These are subjective labels, but after going through many annotations, we collected examples of each. In order to calibrate worker standards to our own, we present curated examples of each in a qualification exam for the evaluation HIT.

Like the qualification exam for the annotation HIT, workers must pass the quiz within three attempts in order to do the real evaluation HIT. If they fail to pass within three attempts, they are automatically blocked from visiting our HITs in the future. Upon passing the quiz, the worker receives $0.30. The reward is lower because this quiz is much simpler. The worker is presented with 20 random meshes, all of which have ground truth labels, and the user must get 75% correct in order to pass the quiz. The ban rate for the evaluation HIT quiz is higher at 14.5%, but this is acceptable because we do not need many workers to evaluate 3D structures; it takes much more effort and time to annotate a structure, than assess if it is correct. It is extremely important that annotation submissions be peer-reviewed correctly.

Upon passing the quiz, the worker reaches the actual evaluation HIT. Each HIT contains a batch of 20 3D structures, 4 of which are sampled from a collection of ground truths for which we have assigned a label. Workers must get 3 out of the 4 ground truth examples in each batch of 20 correct, in order to be approved. This mechanism is necessary because workers can otherwise quickly click through examples and not give much attention to each structure. Workers who fail the ground truth examples are rejected and suffer a penalty to their reputation score. This incentivizes

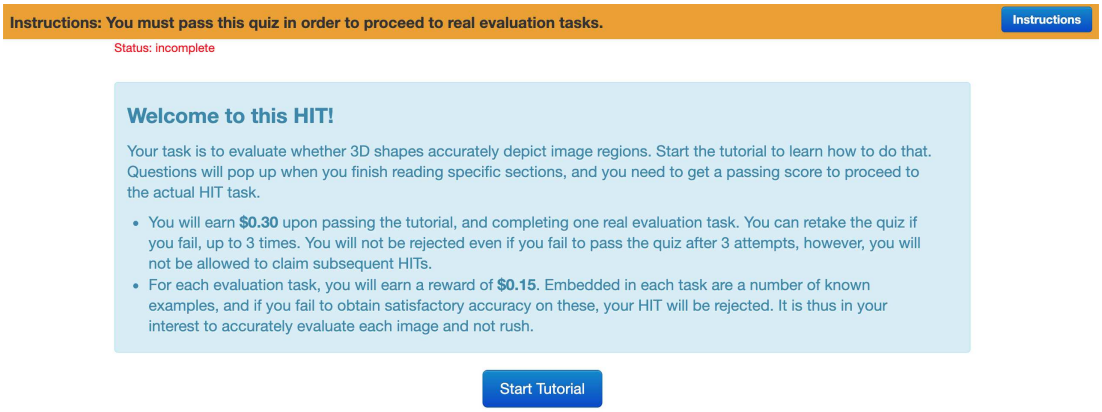


Figure 3.6: Worker qualification quiz for quality evaluation HIT.

the reviewer to be honest in their evaluation of other people's work. Workers who submitted annotations may evaluate other people's submissions, but not their own — for obvious reasons. The worker automatically receives $0.15 if he/she got 3 out of the 4 ground truth examples correct. Reward for the evaluation HIT is fixed, because there is no measure of whether or not an evaluator did a better job than someone else (assuming they both got the ground truth examples right).

This evaluation HIT also ensures that the original annotators are incentivized to submit quality annotations, because annotations which receive too many "incorrect" or "poor" labels get rejected, and the submitter is not compensated. Compensation for annotations scales with both correctness (as determined by this evaluation HIT), and structural complexity, which is discussed in the next section. Figure 3.7 shows the quality evaluation UI.

3.6 Worker Compensation

Amazon Mechanical Turk [4] workers are inherently motivated financially to do a good job; each worker has an associated reputation score that correlates with the quality of their work. Some tasks may have a reputation score threshold, below which workers are not allowed to do the task. Thus, workers with lower reputation score

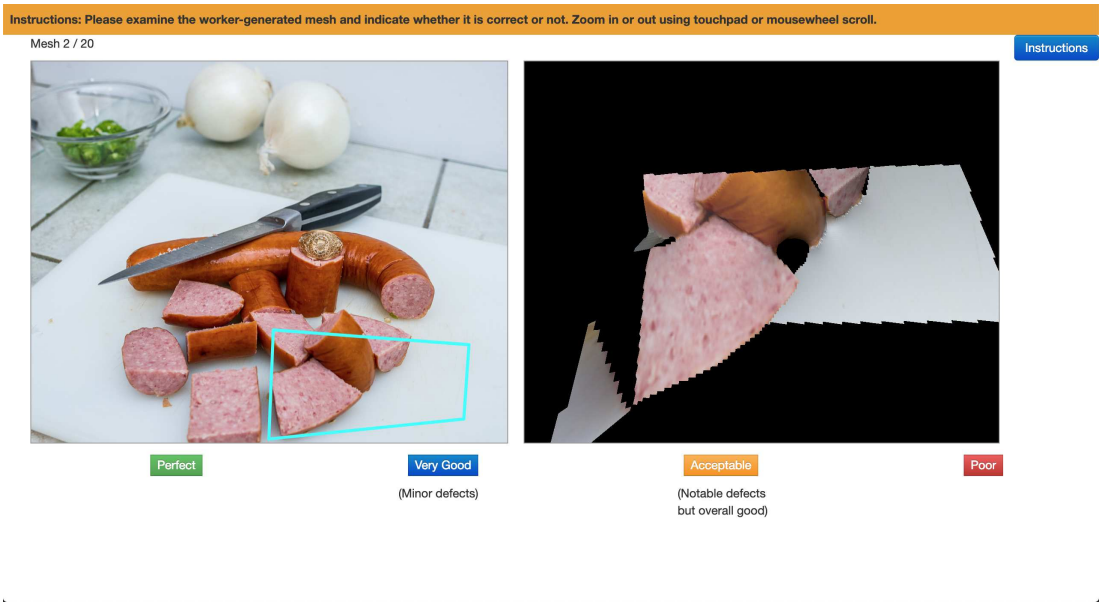


Figure 3.7: Quality evaluation UI.

have strictly fewer opportunities to earn income. Furthermore, for each task that a worker is qualified to do and chooses to do, the task assigner may reject that worker's submissions. The worker only receives reward if his/her submission is approved, and his/her reputation score decreases with rejection.

The incentive disalignment occurs when a task is sufficiently simple for a worker to submit quickly, but the assigner would like a worker to spend more time and do work of higher quality. A worker may choose to "spam" a task, or attempt to do as many tasks as possible for the minimum reward, rather than do a few tasks and earn large bonuses for quality work. The evaluation UI ensures that annotations will be correct; i.e. that the mesh accurately reflects the image region. However, users may learn to optimize for throughput and annotate simple regions without geometric details (e.g. flat walls, bodies of water, etc), that provide little value to our dataset. Thus, we also need a way to quantify structural complexity, and scale compensation accordingly.

3.6.1 Measures of Surface Complexity

Our first attempt to quantify surface complexity was unreliable. The idea was as follows: take a pixel-wise depth map, apply nearest-neighbor downsampling by a factor of x, upsample back to the original resolution using bilinear upsampling, and quantify the difference in depths using the following metric. Let $\hat{d_{xy}}$ be the depth at pixel (x, y) after downsampling and upsampling, and d_{xy} be the original depth at pixel (x, y) before processing. The metric captures scale-invariant depth difference between two images; it is independent of the relative magnitudes of either depth map.

$$\Delta = \sum_{x,y} \frac{(\hat{d_{xy}} - d_{xy})^2}{(\hat{d_{xy}} + d_{xy})^2} \tag{3.1}$$

The intuition behind this approach is that complicated structures should lose more details than simpler structures after downsampling and upsampling. Assuming this to be true, then Δ should be greater for images that are geometrically more complicated. Figure 3.8 demonstrates this qualitatively. However, we found that Δ was not quantitatively greater for more complicated surfaces. We think this is because some images have more background pixels, which lowers Δ because these pixels do not change much during the down/up-sampling process. However, separating background and foreground seemed too troublesome for the simple goal of rewarding workers for investing more effort.

Instead, we chose to use annotation length as a proxy for worker effort. Annotation length also weakly correlates with surface complexity, like our previous approach, but at least it accurately measures worker effort. That is, it strictly takes more effort to annotate 50 points in an image than 10 points.

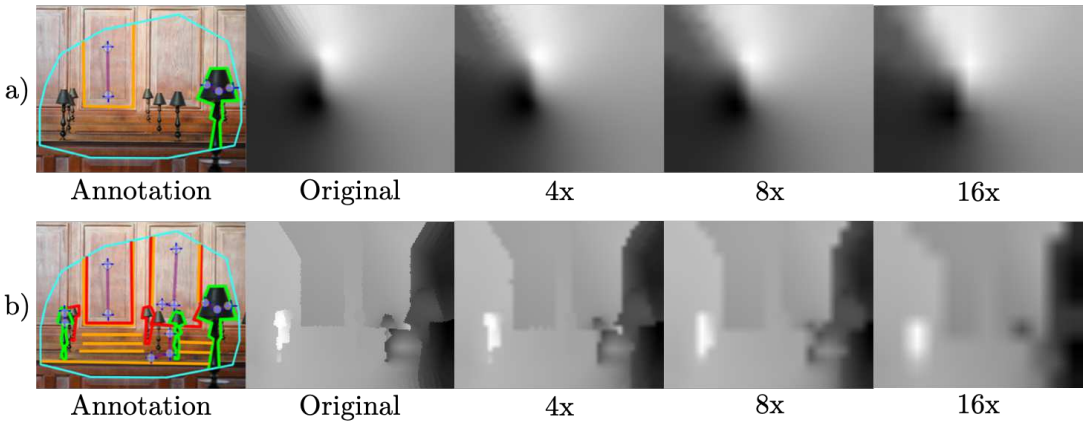


Figure 3.8: The top row is a "simple" annotation, while the bottom row is a "complicated" annotation of the same image. Depth difference after upsampling and downsampling is qualitatively more pronounced in complicated structures.

3.6.2 Annotation Reward Equation

We reward according to the following formula, which computes a quality score (obtained through the evaluation HIT), length score, and then multiplies the two quantities. This compensation formula simultaneously encourages correctness (i.e. whether a 3D reconstruction matches the image region), and effort (as a weak proxy for complexity). We use a threshold of 25 points for the length score, as it is the 80th percentile of annotation lengths, and thus competitive to the average annotation. Note that the total reward is \$0.00 when the quality multiplier is too low. This corresponds exactly to a rejection. If the quality multiplier is above the threshold, the minimum compensation a worker can receive is \$0.10 (i.e. no bonus). A worker can earn at most \$0.70 (i.e. \$0.60 in bonus).

$$
\begin{aligned}
k_{quality} &= \frac{|perfect| + 0.75 * |very\_good| + 0.25 * |acceptable|}{|perfect| + |very\_good| + |acceptable| + |poor|} \\
k_{length} &= \frac{|planar\_normals| + 2 * |curved\_normals| + |fold\_pts| + |occlusion\_pts|}{25} \\
reward &= \mathbb{1}_{(k_{quality} > 0.125)}(0.1 + 0.6 * min(1, k_{quality} * k_{length}))
\end{aligned}
\tag{3.2}
$$

3.7 Dense Surface Normal Computation

Computation of dense surface normals and metric depth, discussed in this section and section 3.8, was done by Shengyi Qian. His work is briefly presented in order to contextualize how we are able to assemble a dense dataset from sparse annotations.

Although the user only annotates sparse surface normals, we are able to recover full surface normal orientation per pixel by solving the below optimization problem. This requires assuming smoothness everywhere on the surface, except at fold and occlusion boundaries. The optimization problem minimizes the sum of squared differences between each normal and its neighbors, excluding fold and occlusion boundaries. Let N_p denote the surface normal at pixel p on a normal map N, and F, O denote pixels belonging to fold and occlusion boundaries. Let $\tilde{N}$ denote the set of user-annotated surface normals at locations P_{known}. Let each pixel p's four neighbors be $\Phi(p)$. If p is on an occlusion boundary, then let the neighbors on the *closer* side of the occlusion be $\Gamma_O(p)$. If p is on a fold line, then let $\Gamma_F(p)$ be all neighbors on a random side of the fold boundary. We then solve for the optimal normal N^* using the Levenberg-Marquandt algorithm [41], and normalize it into a unit vector:

$$N^* = \underset{N}{\operatorname{argmin}} \sum_{p \notin F \cup O} \sum_{\substack{q \in \Phi(p) \\ q \notin F \cup O}} |N_p - N_q|^2 + \sum_{p \in O} \sum_{q \in \Gamma_O(p)} |N_p - N_q|^2 + \sum_{p \in F} \sum_{q \in \Gamma_F(p)} |N_p - N_q|^2$$
$$\text{s.t.} \quad N_p = \tilde{N}_p, \forall p \in P_{known} \tag{3.3}$$

3.8 Dense Depth Computation

Computing depth directly by integrating surface normals fails when surface normals point along the image plane, because the change in depth is infinite. Instead, we assume a perspective camera and construct a linear system based on the perspec-

tive projection of the gradients generated from surface normals. The least-squares solution provides an accurate relative depth map for connected regions, but not between unconnected components. To resolve this ambiguity in global depth ordering, we solve an optimization problem that yields a scaling factor for each surface. The scaling factors force the relative depth between surfaces to obey constraints imposed by occlusion boundaries.

Let the dense depth map after integration be $\mathbf{D}$. We then solve for the scaling factors. Let $\mathbf{S}$ denotes the set of surfaces, and $\mathbf{O}$ denotes the set of occlusion boundaries. Along $\mathbf{O}$, we densely sample a set of point pairs $\mathbf{B} = \{(p, q)\}$. 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 opposite side. The continuous surface a pixel p lies on is $S(p)$, and its depth is D_p. For each $S_k \in \mathbf{S}$, we scale its depth D_{S_k} by X_{S_k}. The set of optimal scaling factors $\mathbf{X}^*$ is solved for in the following optimization problem:

$$\begin{aligned} \mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \quad & \sum_k X_{S_k} \\ \text{s.t.} \quad & X_{S(p)} D_p \geq X_{S(q)} D_q - \Delta, \forall (p, q) \in \mathbf{B} \\ & X_{S_k} > 0, \forall S_k \in \mathbf{S} \end{aligned} \tag{3.4}$$

3.9 Concluding Remarks

We created an end-to-end pipeline for collecting fold boundaries, occlusion boundaries, planarity relationships, and sparse surface normals. From these sparse surface normal ground truths, we are able to compute dense pixel-wise depth and surface normal orientation. Using a relatively small amount of labor, we can obtain pixel-wise ground truths for an arbitrary image. These ground truths exactly define the 3D structure of an object, and were created using human judgment. Can machines

learn to approximate these human-annotated ground truths, and if so, do they then generalize to other datasets? The remaining sections explore this question.

Chapter 4

3D Surfaces in the Wild (3SIW) Dataset

Using our Amazon Mechanical Turk pipeline, we collected ∼27,000 annotations. We call the resultant dataset "3D Surfaces in the Wild", or "3SIW".

4.1 Statistics

- 22,291 approved annotations
- 4,537 rejected annotations
- 621 individual annotators
- 262 individual quality evaluators

Below is a plot of surface normal entropy [16], which quantifies the degree of surface variation. Flat planes have an entropy of zero. A large number of annotations in 3SIW are planar or mostly planar.

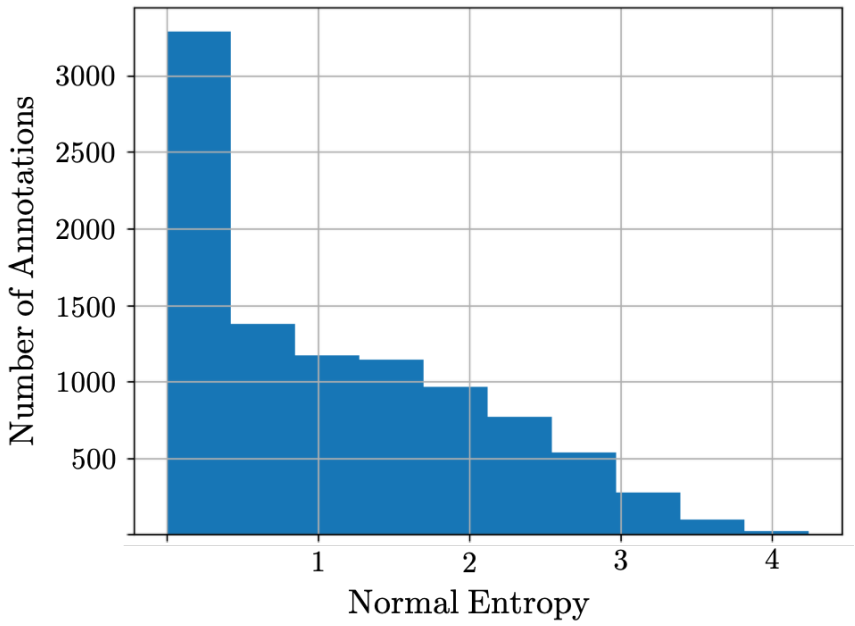


Figure 4.1: Surface normal entropy distribution of 3SIW dataset. Formulation is defined in [16].

4.2 Ground Truths Provided

For each image, the following ground truths are provided:

1. Pixel-wise depth
2. Pixel-wise surface normals
3. Fold and occlusion boundaries
4. Semantic planar segmentation (is a given pixel on a planar or curved surface?)
5. Instance surface segmentation (which surface ID does a given pixel belong to?)
6. Planarity relationships (given two planes, are they parallel, perpendicular, or neither?)

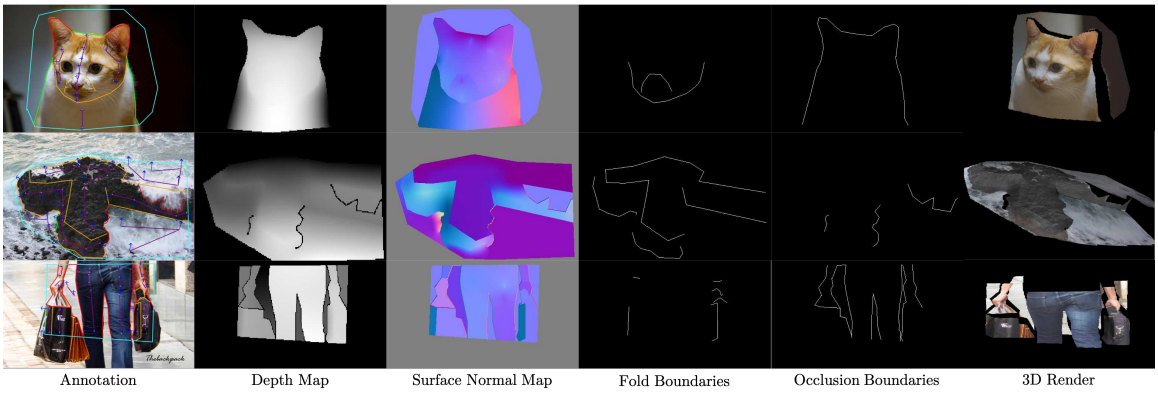


Figure 4.2: Sample visualizations of 3SIW dataset. In the leftmost annotation, orange denotes a fold line, red and green denote occlusions, and the arrows denote surface normal orientation.

Chapter 5

Benchmarks on 3SIW

Our dataset, 3SIW, contains pixel-wise ground truth surface normal orientation and metric depth, exact locations of fold and occlusion boundaries, semantic planar segmentations (i.e. whether or not a pixel belongs to a planar or curved surface), and planarity relationships (i.e. whether or not two planes are parallel, perpendicular, or neither). I present benchmarks for four 3D vision tasks using our dataset, two of which are novel:

1. Surface Normal Estimation
2. Occlusion Detection
3. Fold Detection (novel)
4. Semantic Planar Segmentation (novel)

For each task, we use 3SIW for training and testing state-of-art models, and also report performance on other datasets as a measure of how well our dataset can augment performance on each task. Of the four tasks, only surface normal estimation and occlusion detection are established in the literature. I demonstrate that training on 3SIW advances state-of-art on the Surface Normals in the Wild (SNOW) dataset [10], while training on 3SIW alone leads to nearly state-of-art performance on the

Carnegie Mellon occlusion dataset [50], without training on any data from it. As fold detection and semantic planar segmentation are novel tasks, we report performance on 3SIW only, as a rough upper bound on how well current models can perform on our dataset.

At the start of benchmarking, we only had $14,162$ annotations, and the training/validation/testing split contained $11,329$, 708, and $2,125$ images respectively. We use the same split for all experiments. All results reported here should strictly improve as the dataset grows.

5.1 Surface Normal Estimation

5.1.1 Task Definition

Given a single RGB image, we wish to predict a 3D surface normal map, where each 3D vector corresponds to the x, y, and z components of the surface normal orientation at that pixel. The ground truth is also a 3D surface normal map, and we wish to minimize the total angular difference between our predicted and ground truth output.

5.1.2 Evaluation Metrics

Performance is evaluated on the following five metrics.

1. Mean angle distance (over pixels)
2. Median angle difference (over pixels)
3. Percent of pixels whose angular difference is less than 11.25°
4. Percent of pixels whose angular difference is less than 22.5°
5. Percent of pixels whose angular difference is less than 30°

5.1.3 Methods

Network Architecture

We use an hourglass neural network architecture (Figure 5.1) by Chen et. al [9] that is state-of-art in single-image depth perception, and modify the last layer to output a three channels instead of one. Hourglass networks were first introduced as a state-of-art model for human-pose estimation [42], but have since been used to achieve state-of-art in object detection [13], depth estimation [9], and surface normal estimation [10]. Hourglass networks work by downsampling feature maps at various resolution scales in the first half, and then upsampling to original resolution in the second half. Doing so aggregates features at multiple scales.

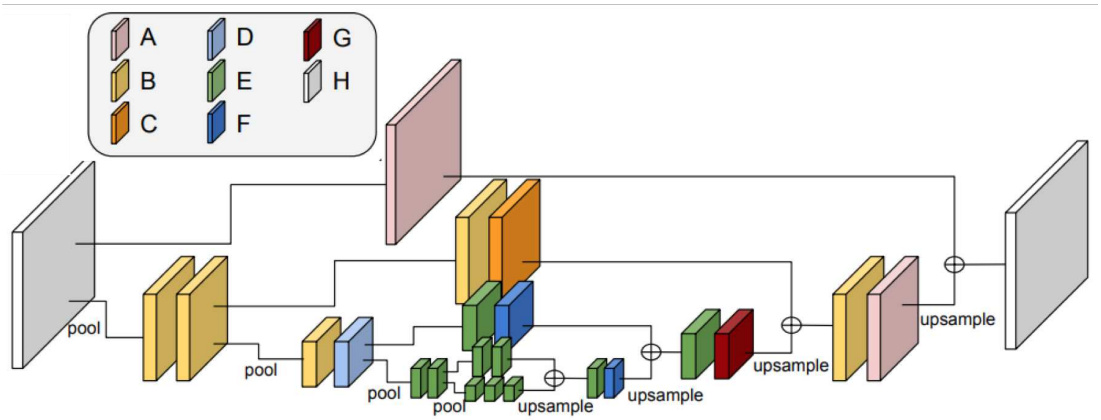


Figure 5.1: Hourglass network architecture. Borrowed from [9].

Surface Normals in the Wild (SNOW) Dataset

SNOW [10] is a dataset for single-image surface normal estimation collected through crowdsourcing. Unlike other surface normal datasets, it is both diverse and large; it contains 60,061 images from the wild. Unlike our dataset, it only provides sparse surface normals; only one pixel per image has a ground truth surface normal. We will demonstrate that training on 3SIW, which provides dense surface normals from

the wild, and then fine-tuning upon SNOW, beats current state-of-art performance on SNOW.

Training

We trained four models: *Our_3SIW*, *Our_3SIW_SNOW*, *Chen_SNOW*, and *Our_SNOW_3SIW*. *Our_3SIW* was trained on only the 3SIW training set for 125,000 iterations with a batch size of 12, learning rate of 10^{-4}, and RMSProp optimizer with no learning rate decay. *Our_3SIW_SNOW* takes *Our_3SIW*, and finetunes upon the SNOW training set for 15,000 iterations using the same hyperparameters. Chen_SNOW was trained on only the SNOW training set for 340,000 iterations with a batch size of 12, learning rate of 10^{-4}, and RMSProp optimizer with no learning rate decay. It represents current state-of-art performance on the SNOW dataset for surface normal estimation. *Our_SNOW_3SIW* takes *Chen_SNOW* and finetunes upon the 3SIW training set for 2,000 iterations using the same hyperparameters.

All models were trained using a simple element-wise loss function that compares the predicted normal at each pixel to the ground truth, using a dot product. Let N and N^* be the predicted and ground truth surface normal maps respectively:

$$L_{normals}(N, N^*) = -\frac{1}{n}\sum_i N_i * N_i^* = -\frac{1}{n} N * N^* \tag{5.1}$$

Models were selected during the training iteration with highest validation performance. Hyperparameters were selected using grid-search.

5.1.4 Results

Our_SNOW_3SIW experiences a large boost in performance on both the SNOW and 3SIW testing set after finetuning on 3SIW, even though the 3SIW dataset is much smaller. This suggests that 3SIW, as a densely annotated dataset, provides more

Model	SNOW					3SIW				
	Angle Distance		Within t°			Angle Distance		Within t°		
	Mean	Median	11.25°	22.5°	30°	Mean	Median	11.25°	22.5°	30°
Our_3SIW	30.22	24.41	22.33	46.83	58.05	28.91	20.45	31.06	53.11	61.90
Chen_SNOW [10]	26.24	20.42	25.60	54.13	67.11	35.10	29.36	15.79	38.60	50.95
Our_3SIW_SNOW	25.13	19.95	**26.64**	**55.71**	**68.68**	28.40	21.94	23.44	51.13	63.58
Our_SNOW_3SIW	**24.84**	**19.83**	26.40	55.51	61.10	**25.14**	**18.73**	**29.36**	**58.12**	**69.57**

Table 5.1: Surface normal estimation performance of different networks on the SNOW [10] and 3SIW test split. Lower is better for the columns labeled "Angle Distance", while higher is better for the columns labeled "Within t°". Our_3SIW_SNOW and Our_SNOW_3SIW both outperform Chen_SNOW, which is state-of-art on the SNOW dataset.

robust training signals than the SNOW dataset, which is sparse. *Our_SNOW_3SIW* outperforms *Our_3SIW_SNOW* on both the 3SIW and SNOW testing set. This may be because 3SIW is still relatively small; with more images, the model should learn more robust and representative priors. Both *Our_3SIW_SNOW* and *Our_SNOW_3SIW* outperform *Chen_SNOW*, attaining new state-of-art performance on the SNOW dataset, demonstrating that training on more diverse data - even in limited quantities - can improve single-image 3D perception and other 3D vision tasks.

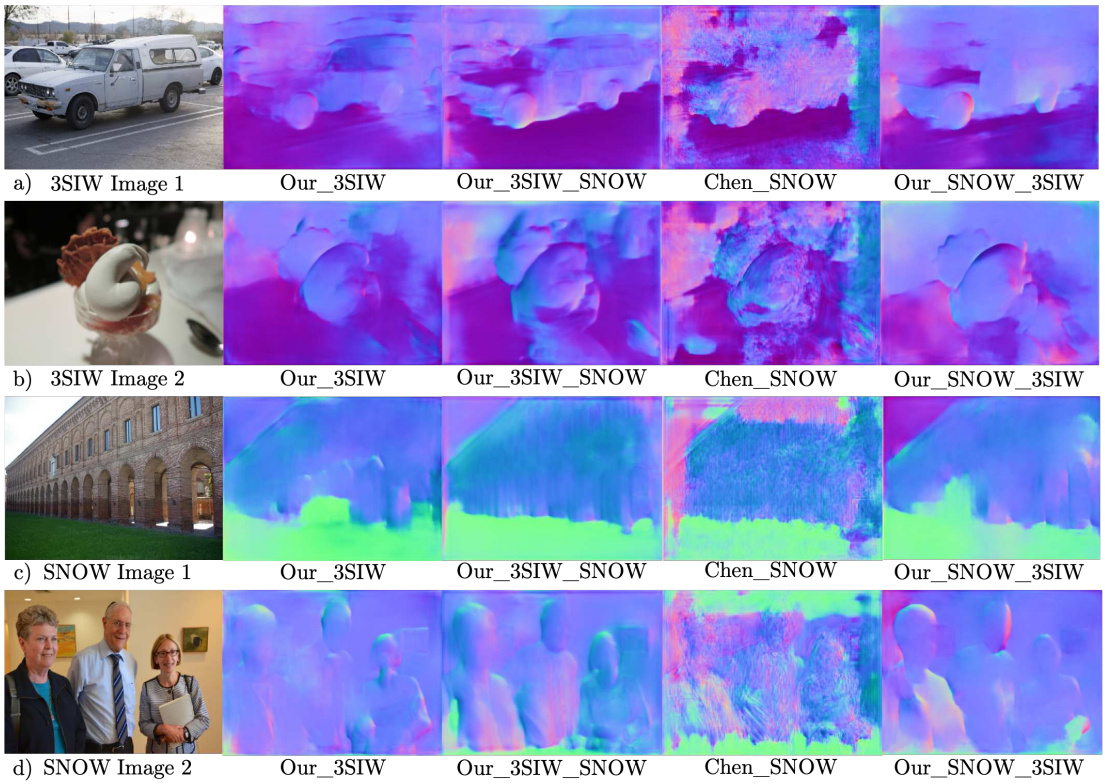


Figure 5.2: Qualitative surface normal predictions on two 3SIW and two SNOW images.

5.2 Occlusion Boundary Detection

5.2.1 Task Definition

Given a single RGB image, we wish to predict a 2D probability map, where each value corresponds to the probability that pixel (x, y) belongs to an occlusion boundary. The ground truth is a 2D binary map indicating whether the pixel belongs to an occlusion boundary.

5.2.2 Evaluation Metrics

Performance is evaluated on the following three metrics.

1. ODS (Optimal Dataset Scale) F-Score

2. OIS (Optimal Image Scale) F-Score

3. AP (Average Precision)

All three metrics are defined in terms of precision and recall, which are two fundamental metrics in binary classification. Precision measures how accurate a classifer's predictions are, or the percent of predictions that were correct. Recall measures how well a clasifier finds all positive instances, or the fraction of all positive instances that were predicted. There is an intrinsic tradeoff between precision and recall; a classifier could be very careful and predict positive only for examples that it is confident about; it would then have high precision and low recall, because all of its positive predictions were correct (true positive), but it incorrectly called other positive examples negative (false negative). Various metrics quantify the tradeoff between precision and recall, and F-score and Average Precision are two such popular metrics.

Let TP be the number of true positives, TN be the number of true negatives, FP be the number of false positives, and FN be the number of false negatives. F-score is defined as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F - score &= 2 * \frac{precision * recall}{precision + recall} \end{aligned} \tag{5.2}$$

ODS and OIS F-score are two variants of F-score defined in the context of probability maps. At a particular threshold $0 < p < 1$, all pixels with probability greater than p are positive, and all pixels with probability less than p are negative. Precision and recall will be different for each threshold, because some pixels may be negative at lower thresholds but positive at higher thresholds. ODS F-score is calculated by taking a fixed threshold $0 < p < 1$, for which the average F-score across all images is maximized. OIS F-score is calculated by taking the threshold $0 < p_i < 1$ for each im-

age i, for which the F-score is maximized. Because OIS F-score changes the threshold per image to maximize each individual image's F-score, while ODS F-score assumes a fixed threshold across the dataset, OIS F-score is never worse than the ODS F-score and often better. Both are popular in the literature.

Average Precision is defined as the integral of the precision-recall curve, formed by plotting precision and recall at multiple probabilities.

5.2.3 Methods

Network Architecture

We use HED (Holistically-Nested Edge Detection) [52], which is a multi-scale convolutional network that computes image features at multiple scales, and then combines all scales using a weighted fusion function. The weights of this fusion function are learned during training. HED is state-of-art on the Berkeley Segmentation Dataset (BSDS) [1], which contains edge boundaries for 500 images. Edges are defined in RGB space; both folds and occlusions are considered edges in the BSDS dataset. Although occlusions and fold boundaries are defined in 3D space, we nevertheless adopt the HED network architecture for our problem formulation.

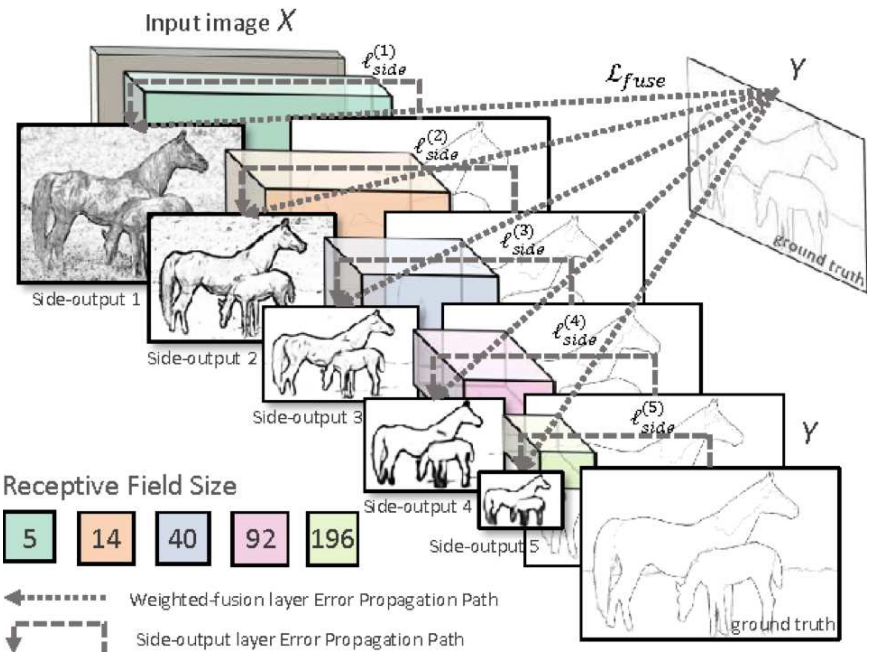


Figure 5.3: HED network architecture. Borrowed from [52].

Berkeley Segmentation (BSDS500) Dataset

BSDS contains manually annotated ground truth edge contours for 500 images. Because edges are defined in RGB space, we do not attempt to improve state-of-art scores on BSDS, like we did in the previous section with SNOW. Instead, we use BSDS as a baseline for how well existing models can expect to do on 3SIW; we test both a BSDS pre-trained model, and a BSDS pre-trained model finetuned upon 3SIW, on 3SIW.

Carnegie Mellon Occlusion Dataset

The CMU occlusion dataset contains manually annotated ground truth occlusion boundaries for 30 videos. Current state-of-art for it is a joint convolutional neural network and conditional random field model by Fu et al. [17]. Because the code for most papers using the CMU dataset is not open-sourced, and the CMU dataset is both small and relatively outdated, we do not train on it. We simply make predictions on the CMU occlusion dataset using 3SIW trained models, as a frame of reference

for how well 3SIW trained models can generalize to other occlusion datasets, without any training whatsoever.

Training

We trained three models for occlusion detection: HED_{BSDS}, $HED_{BSDS\_3SIW\_o}$, and HED_{3SIW_o}. HED_{BSDS} was trained on the BSDS training set for 165,000 iterations with a batch size of 10, base learning rate of 10^{-6}, and a SGD optimizer. Xie et al. [52] alter learning rate for specific layers of the model, and we follow their suggestions for scheduling learning rate decay. $HED_{BSDS\_3SIW\_o}$ takes HED_{BSDS} and finetunes upon the 3SIW training set for 275,000 iterations using a base learning rate of 10^{-9}, but otherwise uses the same hyperparameters. HED_{3SIW_o} was trained on the 3SIW training set for 165,000 iterations with a batch size of 10, base learning rate of 10^{-9}, and the same learning rate scheduler as Xie et al. [52].

All models use a weighted binary cross-entropy loss function and were trained using a SGD optimizer. Models were selected during the training iteration with highest validation performance.

Post-Processing

Before computing the ODS F-score, OIS F-score, and AP calculation code, we first process the network outputs using non-maximal suppression. This is standard practice in edge detection, because non-maximal suppression thins lines and removes noise.

5.2.4 Results

Model	CMU	3SIW		
	OIS	ODS	OIS	AP
HED_{BSDS}	0.629	0.492	0.636	0.426
HED_{3SIW_o}	0.665	**0.617**	0.710	**0.627**
$HED_{BSDS\_3SIW\_o}$	**0.670**	0.613	**0.712**	0.619
Xie et al. (hard) [52]	0.63	-	-	-
Xie et al. (soft) [52]	**0.71**	-	-	-

Table 5.2: Occlusion detection performance of HED networks trained on BSDS and/or 3SIW, and evaluated on the CMU and 3SIW testing set. Higher is better for all metrics.

Training on only 3SIW (HED_{3SIW_o}), as well as finetuning a BSDS-pretrained model on 3SIW ($HED_{BSDS\_3SIW\_o}$), both outperform the second-best score in the literature for the CMU occlusion dataset, which is Xie et al.'s "hard" classifier. This is without any training on the CMU dataset; our 3SIW-trained network is able to achieve near state-of-art performance on the CMU dataset, without previously seeing any CMU images. This suggests that our dataset provides robust training signals for occlusion detection, and generalizes well to zero-shot settings. We do not train on the CMU dataset because it only contains 30 images; reporting results after training on such a small set of images provides little meaning. We would most likely be able to top Xie et al.'s performance upon training on the CMU dataset.

On the 3SIW dataset, HED_{3SIW_o} does better than HED_{BSDS}, as expected. The BSDS dataset is more ambiguous, as it combines folds and occlusions into one semantic category. As we test specifically on the occlusion ground truths, it makes sense that a network trained on 3SIW occlusions would perform better than one trained on

BSDS edges. Finetuning HED_{BSDS} with 3SIW does not noticeably improve performance on the 3SIW testing set.

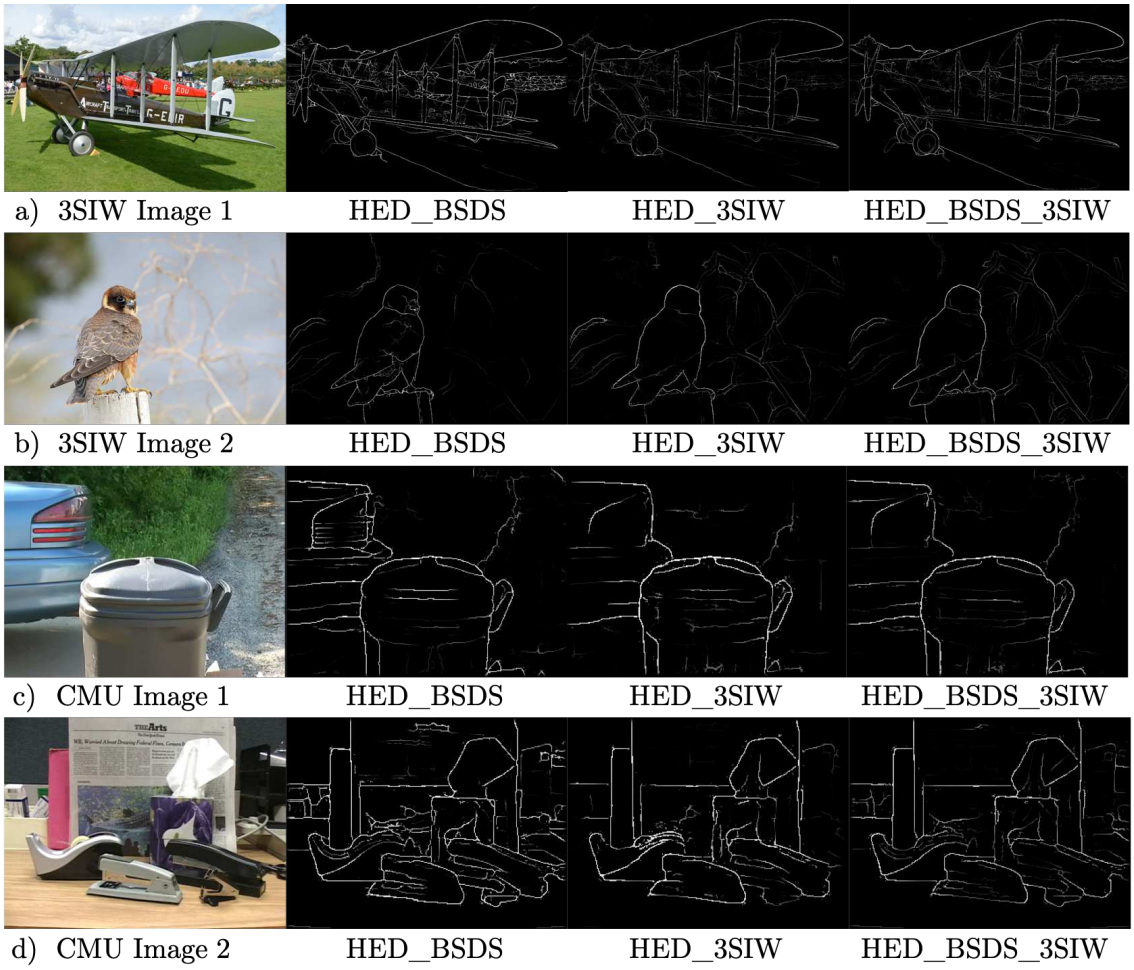


Figure 5.4: Qualitative occlusion boundary predictions on two 3SIW and two CMU images.

5.3 Fold Boundary Detection

5.3.1 Task Definition

Given a single RGB image, we wish to predict a 2D probability map, where each value corresponds to the probability that pixel (x, y) belongs to a fold boundary.

The ground truth is a 2D binary map indicating whether the pixel belongs to a fold boundary.

This task is a novel contribution to the literature, made possible by the heterogeneity of our data. Fold boundaries delineate surface normal discontinuities, or where the surface geometry changes abruptly but remains connected. Learning to estimate fold boundaries, in conjunction with occlusions, provides the ability to reason about physical connectivity and curvature – which is critical for 3D scene understanding [23].

5.3.2 Evaluation Metrics

Performance is evaluated on the same three metrics as in occlusion detection (Section 5.2).

1. ODS (Optimal Dataset Scale) F-Score
2. OIS (Optimal Image Scale) F-Score
3. AP (Average Precision)

5.3.3 Methods

Network Architecture

We use HED (Holistically-Nested Edge Detection) [52], which is a multi-scale convolutional neural network that computes image features at multiple scales, and then combines all scales using a weighted fusion function. The weights of this fusion function are learned during training. HED is state-of-art on the Berkeley Segmentation Dataset (BSDS) [1], which contains edge boundaries for 500 images. Edges are defined in RGB space; both folds and occlusions are considered edges in the BSDS dataset. Although occlusions and fold boundaries are defined in 3D space, we adopt the HED network architecture for our problem formulation.

Berkeley Segmentation (BSDS500) Dataset

BSDS contains manually annotated ground truth edge contours for 500 images. Because edges are defined in RGB space, we do not attempt to improve state-of-art scores on BSDS, as we did in the previous section with SNOW. Instead, we use BSDS as a baseline for how well existing models can expect to do on 3SIW; we test both a BSDS pre-trained model, and a BSDS pre-trained model finetuned upon 3SIW, on 3SIW.

Training

We trained three models for fold detection: HED_{BSDS}, $HED_{BSDS\_3SIW\_f}$, and HED_{3SIW_f}. HED_{BSDS} was trained on the BSDS training set for 165,000 iterations with a batch size of 10, base learning rate of 10^{-6}, and a SGD optimizer. Xie et al. [52] alter learning rate for specific layers of the model, and we follow their suggestions for scheduling learning rate decay. $HED_{BSDS\_3SIW\_f}$ takes HED_{BSDS} and finetunes upon the 3SIW training set for 275,000 iterations using a base learning rate of 10^{-9}, but otherwise uses the same hyperparameters. HED_{3SIW_f} was trained on the 3SIW training set for 165,000 iterations with a batch size of 10, base learning rate of 10^{-9}, and the same learning rate scheduler as Xie et al. [52].

All models use a weighted binary cross-entropy loss function and were trained using a SGD optimizer. Models were selected during the training iteration with highest validation performance.

Post-Processing

Before computing the ODS F-score, OIS F-score, and AP calculation code, we first process the network outputs using non-maximal suppression. This is standard practice in edge detection, because non-maximal suppression thins lines and removes noise.

5.3.4 Results

Model	ODS	OIS	AP
HED_{BSDS}	0.169	0.324	0.084
HED_{3SIW_f}	**0.305**	0.404	**0.217**
$HED_{BSDS\_3SIW\_f}$	0.293	**0.407**	0.205

Table 5.3: Fold detection performance of HED networks trained on BSDS and/or 3SIW, and evaluated on the 3SIW testing set. Higher is better for all metrics.

HED_{3SIW_f} outperforms HED_{BSDS} on 3SIW which is expected, because BSDS provides a noisier training signal; it combines folds and occlusions into one semantic category. As we test specifically on the fold ground truths, it makes sense that a network trained on 3SIW fold would perform better than one trained on BSDS edges. Finetuning HED_{BSDS} with 3SIW does not improve ODS F-score or Average Precision on the 3SIW testing set. As there are no existing datasets for fold detection, we present only benchmarks on the 3SIW dataset, which roughly upper-bounds how well the best current models can perform on 3SIW for fold detection.

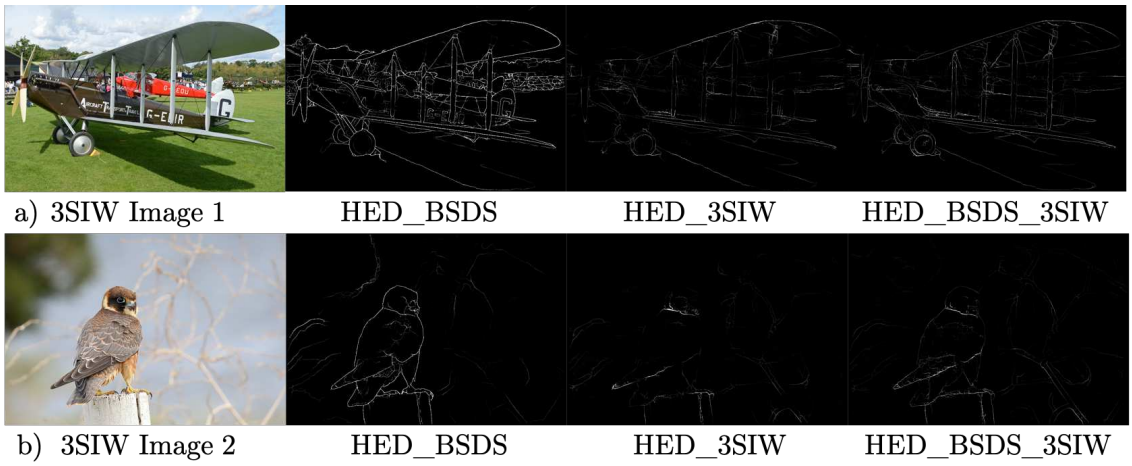


Figure 5.5: Qualitative fold boundary predictions on two 3SIW images.

Compared to the occlusion detection task, performance on fold detection is significantly lower, suggesting that fold detection is a harder task. This is because there are fewer fold boundaries than occlusions in the dataset, and the network tends to overpredict. Thus, performance in fold detection is penalized more-so for false positives.

5.4 Semantic Planar Segmentation

5.4.1 Task Definition

Semantic segmentation is one of the most popular tasks in computer vision. Given a single RGB image, the goal of semantic segmentation is to label each pixel with one of n classes. In our case, $n = 2$, as we wish to classify a pixel as either planar or curved. The ground truth is a 1D class label map indicating whether the pixel belongs to a planar surface, or a curved surface.

This task is a novel contribution to the literature, made possible by the heterogeneity of our data. The ability to recognize whether a surface is planar or not is important, because it allows systems to reason about images on a higher level than individual pixels. Knowing whether a surface is curved or planar imposes additional structural regularity, which leads to a more compact and accurate 3D reconstruction.

5.4.2 Evaluation Metrics

Performance is evaluated on the following two metrics.

1. Class-wise intersection over union (IoU)

2. Mean IoU (mIoU)

Intersection over Union is a popular metric used in object detection, semantic segmentation, instance segmentation, and other computer vision tasks that seek to

maximize overlap between predicted output and the ground truth. In other words, intersection over union measures the number of pixels shared between the predicted output and ground truth labels, divided by the total number of pixels in the output and ground truth. Figure 5.6 demonstrates this visually.

$$IoU = \frac{target \cap prediction}{target \cup prediction} \tag{5.3}$$

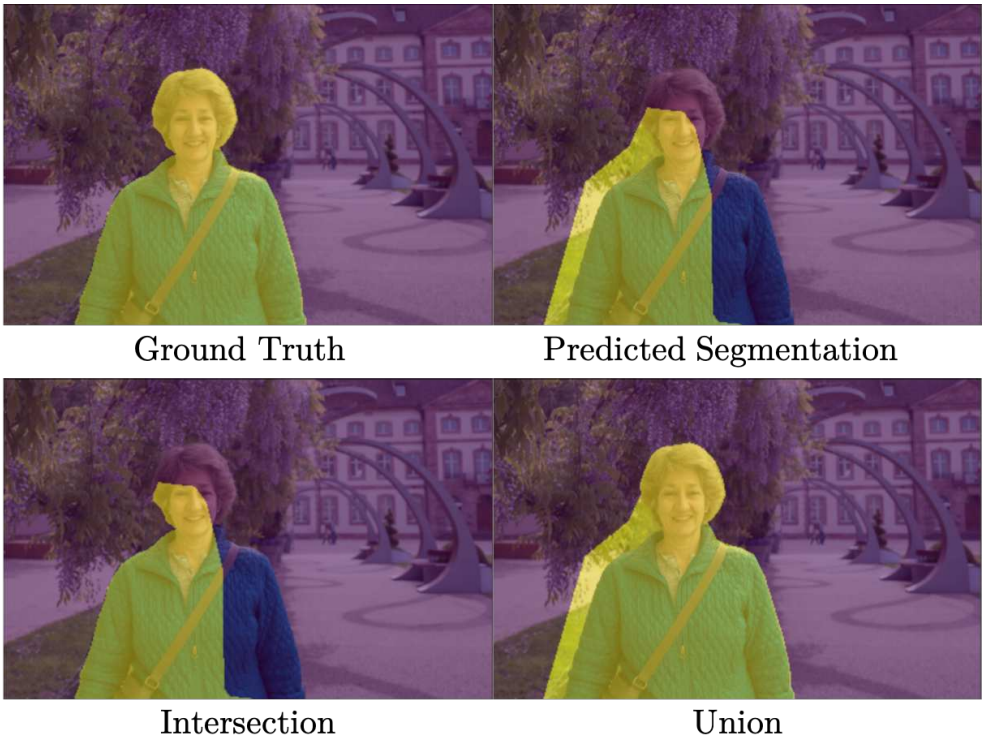


Figure 5.6: Visual demonstration of intersection-over-union calculation. Borrowed from [26].

5.4.3 Methods

Network Architecture

We use DeepLab V3 [6, 7], which is state-of-art in semantic segmentation on the Pascal VOC 2012 challenge [15]. It uses atrous (dilated) convolutions to enlarge

the field of view, and then atrous spatial pyramid pooling to combine convolutional features from multiple scales.

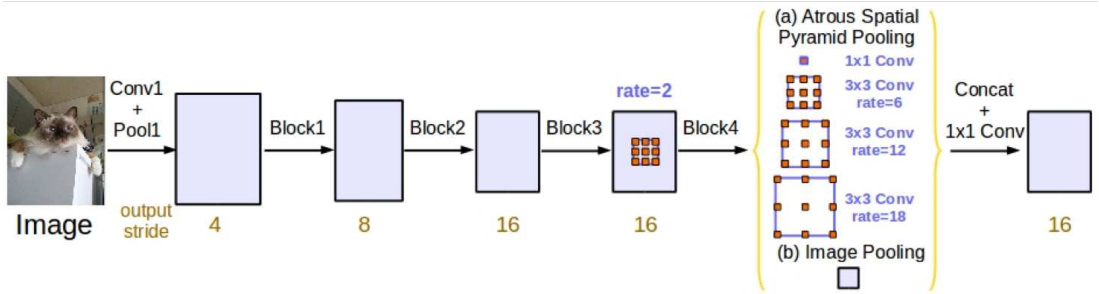


Figure 5.7: DeepLab V3 network architecture. Borrowed from [7].

Training

We trained two models for semantic planar segmentation: $DeepLab_{3SIW\_ce}$ and $DeepLab_{3SIW\_fl}$. $DeepLab_{3SIW\_ce}$ was trained on the 3SIW training set using weighted cross-entropy loss for 15,000 iterations with a batch size of 24, base learning rate of 0.001, and SGD optimizer following the polynomial learning rate suggested in Chen et al. [7] paper. $DeepLab_{3SIW\_fl}$ was trained using focal loss [35] for 15,000 iterations with a batch size of 24, base learning rate of 0.007, and SGD optimizer following the same learning rate scheduler. We decided to experiment with focal loss, as 3SIW is heavily skewed towards planar surfaces. This made it initially difficult to get reasonable IoU for curved surfaces. Focal loss was invented precisely to combat class imbalance for multi-class classification, and we indeed obtained better results using focal loss, compared to cross entropy loss. Focal loss is defined as follows. Let log_{pt} equal the cross entropy loss for a predicted label map and its corresponding ground truth label map. I chose $\gamma = 2$ and $\alpha = 0.5$.

$$focal\_loss = -(1 - e^{log_{pt}})^{\gamma} \alpha * log_{pt} \tag{5.4}$$

Models were selected during the training iteration with highest validation performance. Hyperparameters were selected using grid-search.

5.4.4 Results

Model	Planar IoU	Curved IoU	mIoU
$DeepLab_{3SIW\_ce}$	0.721	0.361	0.541
$DeepLab_{3SIW\_fl}$	**0.749**	**0.472**	**0.610**

Table 5.4: Semantic segmentation performance of the DeepLab network trained on 3SIW using cross-entropy and focal loss respectively, and evaluated on the 3SIW testing set. Higher is better for all metrics.

Planar IoU is significantly higher than curved IoU, however, this makes sense as the dataset skews heavily towards planar (see Figure 4.1). Focal loss is better at countering this class imbalance than cross entropy loss.

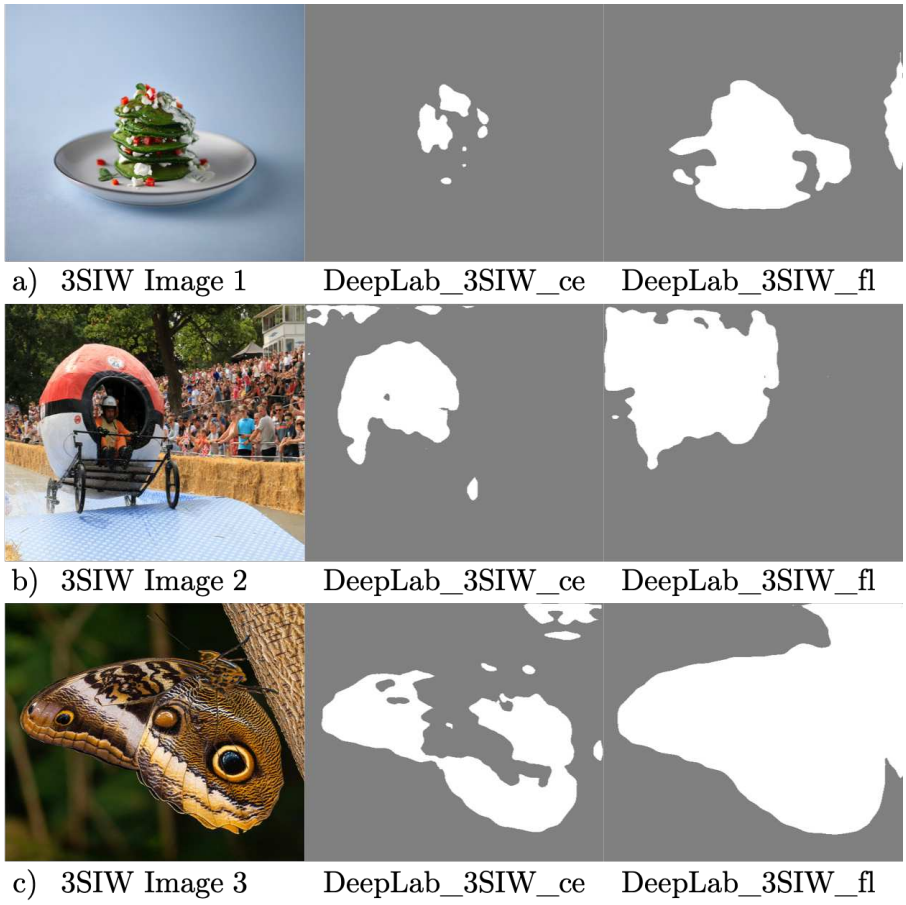


Figure 5.8: Qualitative semantic planar segmentation predictions on three 3SIW images. White denotes curved, and gray denotes planar.

Chapter 6

Conclusion and Future Work

In this work, we addressed existing challenges and limitations of single-view 3D reconstruction, by crowdsourcing a novel large-scale dataset – "3SIW" – for 3D vision. Unlike previous works, we developed an annotation pipeline that is scalable, robust, and recovers dense pixel-wise ground truths from only sparse annotations. Using this efficient data collection method, we assembled 3SIW, which contains 20,000 images and their human-annotated ground truths. Using 3SIW, we then provided benchmarks for four tasks: surface normal estimation, occlusion boundary detection, fold boundary detection, and semantic planar segmentation. The first two tasks exist in the literature, and we demonstrate that training on 3SIW achieves new state-of-art performance on a popular surface normal dataset. We also demonstrate that by training on 3SIW alone, we are able to nearly equal existing state-of-art performance from models trained and evaluated on the Carnegie Mellon Occlusion dataset. The latter two tasks are novel, but essential for improving the ability of visual systems to reason about 3D geometry rigorously. We present benchmarks on these tasks as an approximate upper-bound for how well existing models can perform on our dataset. In sum, our work further advances the argument for why collecting large-scale, representative, and multi-modal data is essential for maximizing model performance.

Despite containing $20,000$ images, 3SIW is still small. We anticipate expanding 3SIW to $50,000$ annotations in the near future, and redoing all benchmark experiments, as the results should improve with more data. In addition, more benchmark experiments are possible. Planarity classification (i.e. given two surfaces, classify whether they are perpendicular, parallel, or neither) is another novel task enabled by the dataset, but I did not have enough time to setup this experiment. I also did not have time to complete the relative and metric depth estimation experiments, which are established in the literature. We would likely obtain new state-of-art results by training on 3SIW, as observed with surface normal estimation in this thesis.

We hope that the release of 3SIW to the research community will enable new directions of research in single-image 3D perception, as well as improve performance on existing 3D vision tasks.

Bibliography

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[2] Harry G Barrow and Jay M Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence*, 17(1-3):75–116, 1981.

[3] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):159, 2014.

[4] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

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

[9] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016.

[10] Weifeng Chen, Donglai Xiang, and Jia Deng. Surface normals in the wild. In *Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy*, pages 22–29, 2017.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas A Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. *arXiv preprint arXiv:1904.08189*, 2019.

[14] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[16] Torsten Fiolka, Jörg Stückler, Dominik A Klein, Dirk Schulz, and Sven Behnke. Sure: Surface entropy for distinctive 3d features. In *International Conference on Spatial Cognition*, pages 74–93. Springer, 2012.

[17] Huan Fu, Chaohui Wang, Dacheng Tao, and Michael J Black. Occlusion boundary detection via deep exploration of context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 241–250, 2016.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[20] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 449. ACM, 2018.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[24] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[25] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. *arXiv preprint arXiv:1808.01838*, 2018.

[26] Jeremy Jordan. Evaluating image segmentation models, 2018.

[27] Kevin Karsch, Zicheng Liao, Jason Rock, Jonathan T Barron, and Derek Hoiem. Boundary cues for 3d object shape recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2163–2170, 2013.

[28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3, 2017.

[29] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[31] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[32] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[33] Jun Li, Reinhard Klein, and Angela Yao. Learning fine-scaled depth maps from single rgb images. *arXiv preprint*, 2016.

[34] Zhengqi Li and Noah 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.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[37] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[38] David G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987.

[39] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas 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.

[40] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv preprint arXiv:1612.05079*, 2016.

[41] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[42] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[43] Emmanuel Prados and Olivier Faugeras. Shape from shading. In *Handbook of mathematical models in computer vision*, pages 375–388. Springer, 2006.

[44] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[48] Pawan Sinha and Edward Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *1993 (4th) International Conference on Computer Vision*, pages 156–163. IEEE, 1993.

[49] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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

[51] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015.

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

[53] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1281–1290, 2017.

[54] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[56] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. *arXiv preprint arXiv:1808.01454*, 2018.