arXiv:2008.03286v1 [cs.CV] 7 Aug 2020

HoliCity: A City-Scale Data Platform for Learning Holistic 3D Structures

Yichao Zhou*[1], Jingwei Huang[2], Xili Dai[1], Linjie Luo[3], Zhili Chen[3], and Yi Ma[1]

[1]University of California, Berkeley, California, USA
[2]Stanford University, Stanford, California, USA
[3]Bytedance Research, Palo Alto, California, USA

Abstract

We present HoliCity, a city-scale 3D dataset with rich structural information. Currently, this dataset has 6,300 real-world panoramas of resolution 13312×6656 *that are accurately aligned with the CAD model of downtown London with an area of more than 20 km*^{2}*, in which the median reprojection error of the alignment of an average image is less than half a degree. This dataset aims to be an all-in-one data platform for research of learning abstracted high-level holistic 3D structures that can be derived from city CAD models, e.g., corners, lines, wireframes, planes, and cuboids, with the ultimate goal of supporting real-world applications including city-scale reconstruction, localization, mapping, and augmented reality. The accurate alignment of the 3D CAD models and panoramas also benefits low-level 3D vision tasks such as surface normal estimation, as the surface normal extracted from previous LiDAR-based datasets is often noisy. We conduct experiments to demonstrate the applications of HoliCity, such as predicting surface segmentation, normal maps, depth maps, and vanishing points, as well as test the generalizability of methods trained on HoliCity and other related datasets. HoliCity is available at* https://holicity.io.

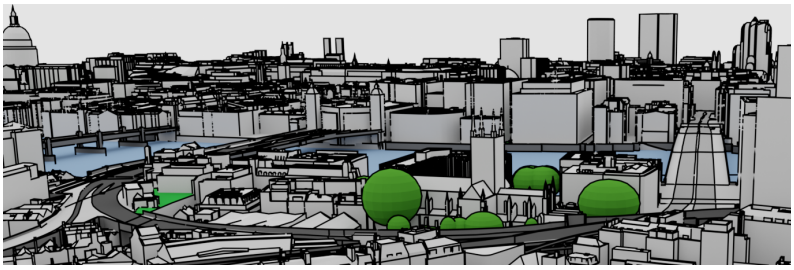
1. Introduction

In the past decades, we have witnessed an increasing demand of 3D vision technologies. With the development of robust point features such as SIFT [34] and ORB [41], structure-from-motion (SfM) and simultaneous localization and mapping (SLAM) have been successfully applied to tasks such as autonomous driving, robotics, and augmented reality. Leading 3D vision products, such as Hololens, Magic Leap, Apple ARkit, Google AR navigation can localize themselves and reconstruct the environment with point clouds. Although the robustness of SfM has been greatly improved over the past decades, the resulting point clouds are still noisy, incomplete, and thus can hardly be directly used in real-world applications. Intricate post-processing procedures, such as plane fitting [23], Poisson surface reconstruction [26], and TSDF fusion [13] are necessary for downstream applications. Increasingly have people found that these long pipelines of 3D reconstruction are difficult to implement correctly and efficiently, and results in low-level representations such as point clouds are also unfriendly for parsing, editing, processing, and visualization.

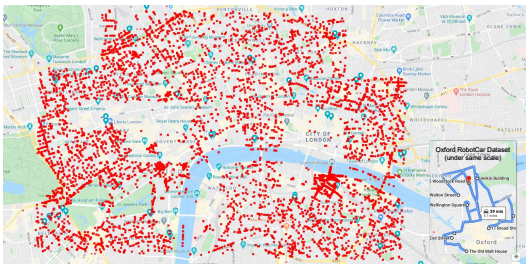
Looking back at the origin of computer vision from the '80s, researchers have found that our human beings do not perceive the world with SIFT-like point features [12, 61, 38]. Instead, we abstract scenes with high-level geometry primitives, such as corners, line segments, and planes, to form our sense of 3D, navigate in cities, or interact with environments. This hints us that instead of point clouds, we can also use high-level structures as a representation for 3D reconstruction, which in many cases are more compact, intuitive, and easy to process. In fact, early 3D vision research does focus on reconstructing shapes with high-level abstractions, such as lines/wireframes [17, 53, 6], contours/boundaries [51, 29], planes/surfaces[3, 62], and cuboids/polyhedrons [45, 35, 52, 59, 25, 53, 60, 2, 48]. We name these high-level abstractions *holistic structures* in this paper, as they tend to represent scenes globally, comparing to the SIFT-like local features. However, recognition of holistic structures from images seems too challenging to be practical at that time. 3D reconstruction with high-level abstractions does not get enough attention despite its potentials, until recently.

Inspired by the recent success of deep convolutional neural networks (CNNs) in image classification, researchers have proposed a variety of neural network-based approaches to extract high-level holistic structures from images, such as wireframes [73, 72], planes [31, 33], cuboids [40], vanishing points [71], room layouts [74], and building layouts [70]. Most of them are supervised learning algorithms, which means that they rely on datasets with annotated holistic structures for training. However, making a properly an-

*This work is sponsored by a generous grant from Sony Research US.



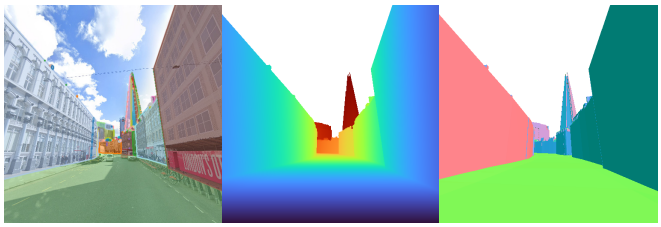
(a) Bird's-eye view of the HoliCity CAD model



(b) Viewpoint coverage

(c) Panorama

(d) RGB



(e) Renderings (planes, depth, normal)

Figure 1: Our HoliCity dataset consists of accurate city-scale CAD models and spatially-registered street view panoramas. HoliCity covers an area of more than 20 km^2 in London from 6,300 viewpoints, which dwarfs previous datasets such as Oxford RobotCar [37] (1b). From the CAD models (1a) and the panoramas (1c), it is possible to generate all kinds of clean structured ground-truths for 3D scene understanding tasks, including perspective RGB images (1d), depth maps, plane maps, and normal maps (1e).

notated outdoor 3D dataset for a particular high-level representation is complex. The building process usually has 2 stages: (1) 3D data collection and (2) structure labeling. Collecting 3D data such as depth images is a cost- and labor-intensive process. This is especially true for outdoor scenes due to the lack of dense depth sensors. Even with expensive LiDAR systems, the point clouds from scans are still noisy and have lower spatial resolution compared to RGB images. Derived features such as surface normal are unsmooth, which might be the reason why previous normal estimation research [16, 4, 58, 24] only demonstrates their results on indoor scenes. These characteristics are unfavourable for extracting holistic structures. In addition, labeling high-level abstractions on the collected 3D data is also challenging. On one hand, manually annotating high-level structures is time-consuming, as it requires researchers to design complicated labeling software in 3D, train annotators to label the data with a unified standard, and do the quality control. On the other hand, the quality of automatically extracted holistic structures from fitting algorithms such as J-Linkage [54] might not be adequate. The results can be inaccurate, incomplete, and erroneous, especially when the quality of 3D data is not that good and the supporting features of the high-level structure are small. To make the problem worse, frequently a dataset that is labeled for one particular structure cannot be easily reused for other structures. As a result, data preparation has become one of the major road blockers for structural 3D vision research.

To address the aforementioned challenges and provide a high-quality multi-purpose dataset for the vision community, we develop HoliCity as a data platform for learning holistic 3D structures in urban environments. Figure 1 shows the illustration of HoliCity. HoliCity is composed of 6,300 high-resolution real-world panoramas that are accurately aligned with the 3D CAD model of downtown London with more than 20 km^2 of area (see Figures 1a to 1c). Instead of relying on expensive vehicle-mounted LiDAR scanners, HoliCity takes the advantage of existing high-quality 3D CAD city models from the GIS community. This way, we can collect a large area of 3D data with fine details and semantic labels at relatively low cost, in which the CAD models are parametrized by corners, lines, and smooth surfaces so that it is friendly for researchers to extract holistic structures. In comparison, traditional LiDAR-based datasets such as KITTI [20] and RobotCar [37] cover a much smaller area (see Figure 1b for visual comparison), are more expensive to collect, and use noisy point clouds as their representation. Furthermore, the panorama photographs in HoliCity are sharp, professional captured, and with resolution as high as 13312×6656. In contrast, images of LiDAR-based datasets are often from video recordings, so they can be blurry, low-resolution, and repetitive. Application-wise, traditional LiDAR-based datasets focus on tasks related to low-level representations, such as depth map prediction, re-

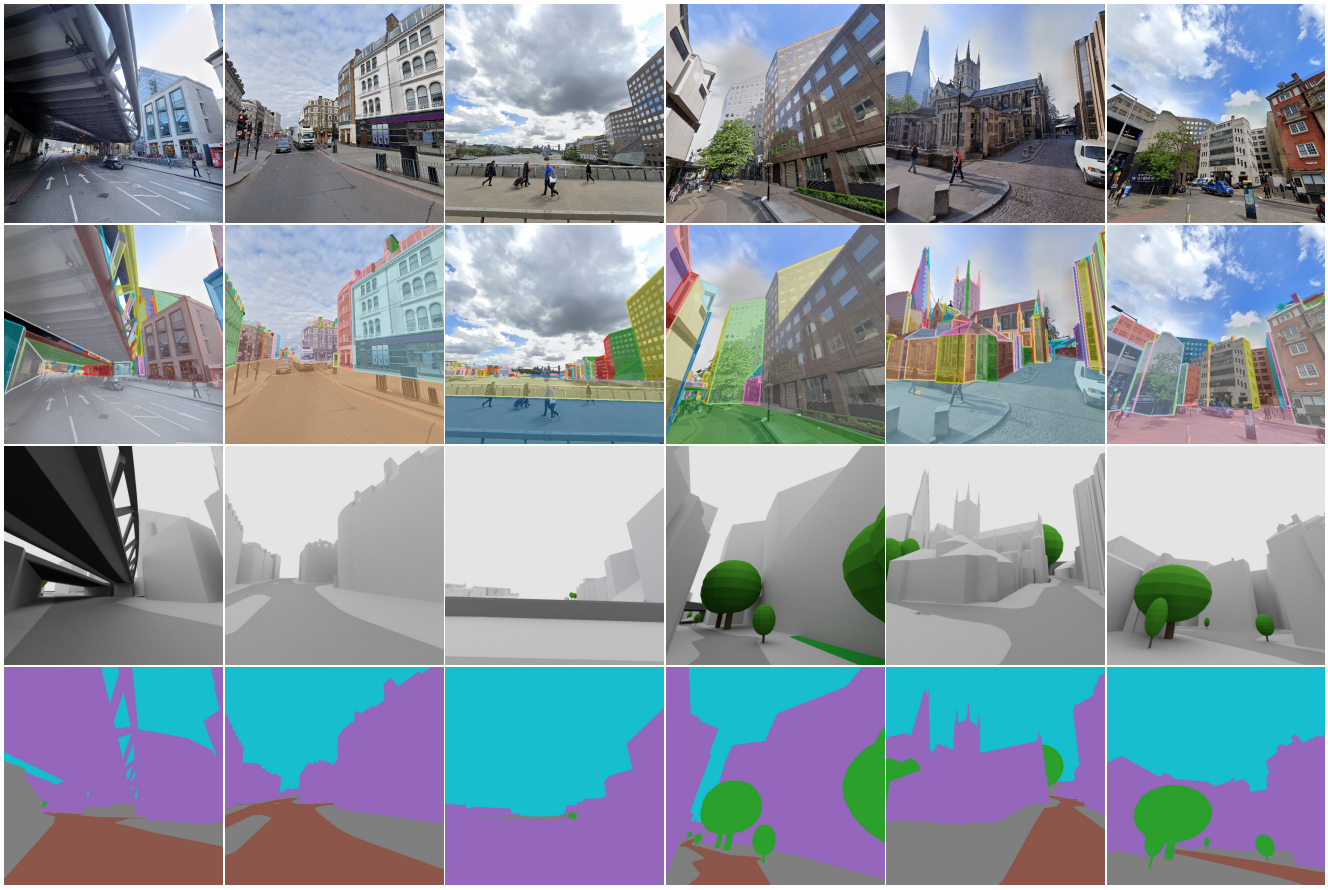


Figure 2: Images and generated 3D information from sampled viewpoints of HoliCity dataset. From top to bottom: perspective images rendered from panoramas, surface segments overlaid with images, CAD model renderings, and semantic segmentation.

construction with point clouds, and camera relocalization, while HoliCity is designed for supporting the research of 3D reconstruction with high-level holistic structures, such as junctions, lines, wireframes, planes, parameterized surfaces, and other geometry primitives that they can be derived from CAD models, in addition to the traditional low-level representations.

In summary, the main contributions of this work include:

1. we propose a novel pipeline for creating a city-scale 3D dataset by utilizing existing CAD models and street-view imagery at a relatively low cost;
2. we develop HoliCity as a data platform for learning holistic 3D structures in urban environments;
3. we accurately align the panorama images with the CAD models, in which the median reprojection error is less than half a degree for an average image;
4. we conduct experiments to justify the necessity of a CAD model-based data platform for 3D vision research, including demonstrating potential applications and testing its generalizability from/to other datasets.

2. Related Work

Synthetic 3D Datasets. Recently, object-level synthetic datasets such as ShapeNet [10] are popular for computer vision research, as people are free to convert 3D CAD models to any representations that their learning-based algorithms like, such as depth maps [11], meshes [21], voxels [66], point clouds [18], and signed distance fields [43]. With the availability of CAD models, not only HoliCity shares similar freedom as these synthetic object-level datasets, but it also offers scene-level real-world images in urban environments. Additionally, synthetic approaches have also been used to create structured 3D scenes, as seen in SceneNet [39], SUNCG [50], SYNTHIA [47] and GTA5 [44] datasets. They provide perfect labels for depth information and semantic segmentation, and it is also possible to extract high-level structural information from them. Nevertheless, their images are still fake. In our experiments, we find that there exists a large domain gap between the virtual renderings of synthetic datasets and our real-world images.

	NYUv2	ScanNet	Stanford-2D-3D	SYNTHIA	MegaDepth	KITTI	RobotCar	**HoliCity**
type	real	real	real	synthetic	real	real	real	**real**
scene	indoor	indoor	indoor	driving	landmark	driving	driving	**city**
depth	RGBD	RGBD	RGBD	CAD	SfM	LIDAR	LIDAR	**CAD**
style	dense	dense	◯	dense	dense	quasi	quasi	**dense**
normal	◯	✓	✓	◯	◯	◯	◯	✓
plane	◯	✓	◯	✓	◯	◯	◯	✓
coverage	/	0.034 km^2	0.006 km^2	/	/	/	/	**20 km^2**
path length	/	/	/	/	/	39.2 km	10 km	**/**
time span	1 scan	1 scan	1 scan	/	unknown	5 scans	2014-2015	**2008-2019**
diversity	464 rooms	707 rooms	4 buildings	/	200 scenes	path	path	**city**
# of images	1.4k	2.5m	1.4k	50k	100k	93k	20m	**6.3k**
source	image	video	video	/	image	video	video	**image**
FoVs	$71° \times 60°$	$45° \times 34°$	panorama	$100° \times 84°$	random	$90° \times 35°$	multi-cam	**panorama**
sementics	2D	3D	3D	3D	N.A.	N.A.	N.A.	**3D**
scale	absolute	absolute	absolute	absolute	relative	absolute	absolute	**absolute**
max depth	(indoor)	(indoor)	(indoor)	∞	(relative)	80m	50m	∞

Table 1: Comparing HoliCity with existing datasets for 3D reconstruction and scene understanding. We list the features of NYUv2 [42], ScanNet[14], Stanford-2D-3D-Semantics [1], SYNTHIA [47], MegaDepth [30], KITTI [20], and RobotCar [37]. The ◯ in the normal and plane rows represents that it might be possible to use fitting algorithms such as J-Linkage [54] to get the annotations, but the quality might suffer due to the noise in point clouds.

Outdoor Datasets. Due to the high cost and the limitation of LiDAR systems, acquiring 3D measurements for outdoor scenes is difficult. Publicly available datasets created with LiDAR technology, such as KITTI [20] and RobotCar [37], have a relatively small scale and low spatial resolution, and mainly focus on the driving scenarios where the camera is facing toward the road. Recently, more outdoor datasets emerge by leveraging structure-from-motion (SfM) and multi-view stereo (MVS) on web imagery in-the-wild, such as MegaDepth [30], and Web Stereo Video Dataset [57]. These datasets provide depth information at a low cost with the expense of quality, because visual 3D reconstruction is not really accurate or robust for random Internet images. In addition, previous 3D outdoor datasets mainly use point clouds as their representation, which are usually noisy. Hardly any of them provide structured ground-truths such as lines, wireframes, segmented 3D planes, and identified buildings for structured urban scene understanding. In comparison, HoliCity offers high-quality CAD models and ground truth of holistic 3D structures that cover an unprecedented range of areas and viewpoints at the scale of a city (Figures 1 and 2).

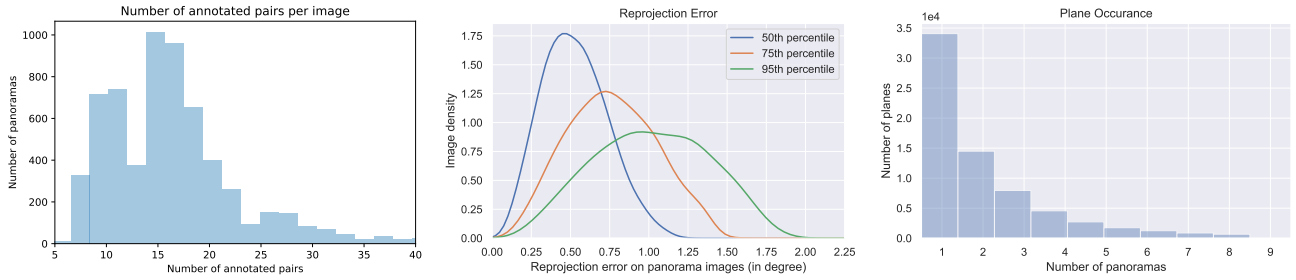
Indoor Datasets. Thanks to increasingly affordable indoor dense depth sensors such as Kinect and RealSense, high-quality real-world indoor 3D data can be produced on a massive scale. Datasets like NYUv2 [42] provide RGBD images for a variety of indoor scenes. Recent datasets such as SUN3D [65], ScanNet [14], Stanford-2D-3D-Semantics [1], and Matterport3D [9] provide surface reconstruction results and 3D sementics annotation in addition to depth maps. The quality of indoor datasets often varies from scenes to scenes, depending on how well the scene is scanned. Compared to HoliCity that provides accurate CAD models designed for learning holistic structures, the noises, holes, and misalignments in the point clouds of these indoor datasets make them not ideal for extracting high-level 3D abstractions. More importantly, our experiment shows that it is unlikely that a network can generalize from indoor training data to outdoor 3D tasks, due to the significant domain gaps.

3. Exploring HoliCity

Our goal is to develop a large-scale outdoor 3D dataset that is rich of holistic structural information. To this end, HoliCity uses commercially available CAD models provided by AccuCities[1], which are reconstructed and built using photogrammetry from high-resolution aerial imagery, each with accurately-recorded GPS position, height, tilt, pitch, and roll. Aerial photogrammetry is a mature technique and it has been widely used to build models with different levels of details for city planning in the field of geographic information systems (GIS). As a result, we are able to get the CAD models that cover a wide range of city areas. The CAD models we use contain details of building features with up to 15 cm accuracy, according to the provider.

To make the CAD models useful for image-based tasks, we need to precisely align the CAD models with images taken from the ground. To do that, we collect the panorama images from Google Street View that cover the same area in

[1]https://www.accucities.com/



(a) Annotations per Image (b) Reprojection Error (c) Plane Occurrence

Figure 3: Statistics of the HoliCity dataset. We show the number of annotations per panorama image to register it to the CAD model (3a); the reprojection errors of annotated 3D points on panoramas (3b) and the occurrence of planes on different panoramas (3c).

the city as the CAD models, along with their geotags. To further increase the alignment accuracy between the panoramas and the CAD models, we implement an annotation software for fine-tuning the geolocation of panoramas to precisely localize them to the CAD models (See Section 4 for technical details).

Table 1 summarizes the difference between our dataset and other existing datasets. Compared to the previous 3D outdoor datasets, HoliCity has its advantage on the following aspects:

Holistic Structures. With CAD models, it is straightforward to extract high-quality holistic structures such as corners, lines, planes, and even curved surfaces from HoliCity compared to the point clouds, as shown in the second row of Figure 2. HoliCity also supports traditional low-level representations such as depth maps and normal maps (Figure 1e), as well as rendering the maps of semantics annotations of roads, buildings, curbs, sky, water, and others, which have already been annotated in the CAD model. In contrast, most existing outdoor datasets use point clouds as the representation. Due to the limitation of LiDAR technology and costs, the point clouds are often too sparse and too noisy for an algorithm to extract such high-level structures reliably. Hardly any of existing outdoor datasets provide high-level structure annotations such as lines, wireframes, segmented surfaces, and identified buildings. The ground truth semantic segmentation also needs to be labeled manually afterward [14, 9].

Coverage. Compared to other datasets, HoliCity is able to cover a much larger area of more than 20 km^2 in downtown London with more diverse urban scenes and viewpoints, thanks to the existing CAD models and street-view panoramas. Figure 1b shows the coverage map that is aligned with Google Maps and compares it against the Oxford RobotCar dataset. HoliCity contains 6,300 panorama images from diverse viewpoints. We note that it might look like that our dataset has fewer images than other datasets, it is actually fairly large among panorama-based ones, such as Stanford-2D-3D [1] (1,413 images), and SUN3D [65] (6,161 images). For the datasets in Table 1 with much higher image counts, their images are mostly extracted from videos, which are highly repetitive and blurry. Therefore, we think "coverage" is a more fair metric for evaluating the size and variety of a dataset, especially considering that our dataset already have had a reasonable density of viewpoints as seen in Figure 1b.

Accuracy. We carefully align the panoramas with the CAD model using a reasonable number of annotated correspondence points between them, as shown in Figure 3a. This is because the original geolocation of Google Street View images is not precise, so we re-estimate the camera pose by minimizing the reprojection error of our annotations. Figure 3b shows the reprojection error of annotated points between the images and the CAD model. We find that for an average image, the median reprojection error is less than half a degree and the 95th percentile does not exceed 1.2 degrees. Besides the accuracy of camera registration, our CAD model-based dataset does not have constraints on maximum depth, unlike the depth obtained from LiDAR. Hence it might be more suitable for evaluating image-based 3D reconstruction algorithms.

Panorama. HoliCity uses the panorama images from Google Street View with resolution 13312×6656. This way, our dataset can capture the full view from each viewpoint and it is not biased towards any directions or landmarks. It also gives us extra flexibility to render many times more perspective images and emulate cameras of different types. In contrast, images from previous outdoor datasets are mainly captured by the front-facing cameras that are towards roads. The field of views (FoVs) is limited and the area of interest is biased.

Multi-view. The number of occurrences of each 3D plane in our panorama database is shown in Figure 3c. More

than half of the planes occur in more than one image and about a third of planes occur in more than two images. This means that our dataset can support the 3D vision research that requires multi-view correspondence between images, e.g., structure-from-motion, multi-view stereopsis, and neural renderings.

Time Span. Most of existing 3D outdoor datasets are collected in short periods of time, as shown in the row "time span" of Table 1. In contrast, HoliCity utilizes the panorama images from Google Street View, which are captured during a span of over 10 years. This greatly increases the variety of data, which can benefit learning-based methods and bring additional challenges to tasks.

4. Building HoliCity

4.1. City Data Collection

3D Models. Although there exist many public city CAD models from the GIS community [28, 55] and municipality governments[2], determining their quality is hard as these datasets are built for different purposes. In this project, we use the commercially available CAD model from AccuCities. Their CAD model covers the area of downtown London and comes with two levels of details. The low-resolution version (cover 20 km^2) has details accurate to 2m, while the high-resolution version (covers 4 km^2) are accurate to 15cm in all three axes. The CAD model is stored in the mesh format and each surface is tagged with semantic types such as `BUILDING`, `TERRAIN`, `BRIDGE`, `TREE`, etc.

Street-View Images. We collect street-view panorama images from Google Street View. At each viewpoint, we have a 360° panorama image along with the geographic data of the camera from GPS and IMUs: 1) latitude and longitude in WGS84 coordinate; 2) azimuth, the angle between the forward-up plane of the camera and geographic north; 3) a unit vector representing the up direction of the camera with respect to the direction of gravity. The geographic information along from Google Street View is not sufficient for accurately registering the camera pose between the CAD model and the panorama images. First, we do not have the elevation of the camera. We estimate the initial z of the camera by adding 2.5m (the height of the camera on the car) to the ground elevation, as the terrain is provided in our CAD model. Second, the provided GPS and IMU data are not accurate enough for a decent alignment between the panorama images and the CAD model. Therefore, we resort to human annotation for registration.

4.2. Annotation Pipeline

Our annotation pipeline contains two steps: 1) registering the CAD model with the WGS84 coordinate by annotating key points on Google Maps and CAD models; 2) fine-tuning the registration by labeling the 2D-3D correspondence between the vertices of the CAD model and the pixels of the panorama.

Geotagging the CAD Models. In the first step, we register the CAD model with the WGS84 coordinate used by Google Street View. To do that, we annotate 44 corresponding 2D locations on both Google Maps and our CAD model. We label most of the points on the inner corners of roof ridges to maximize the registration accuracy. We employ a nonlinear mesh deformation model for registration. Let $\mathbf{X}_{\text{WGS}}$ and $\mathbf{X}_{\text{CAD}}$ be the 2D coordinates of the points on Google Maps and our CAD models and Γ be the mapping from $\mathbf{X}_{\text{CAD}}$ to $\mathbf{X}_{\text{WGS}}$ parameterized by Ω. Mathematically, we have

$$\Gamma(\mathbf{X}_{\text{CAD}}, \Omega) = \mathbf{X}_{\text{WGS}} \tag{1}$$

Here, we use $\Omega[x, y] \in \mathbb{R}^2$ is a 2D lookup table and Γ simply bilinear interpolates Ω and returns $\Omega[\mathbf{X}_{\text{CAD}}]$. We can find the optimal $\hat{\Omega}$ by optimizing

$$\min \|\Gamma(\mathbf{X}_{\text{CAD}}, \Omega) - \mathbf{X}_{\text{WGS}}\|_2^2 + \lambda \|\Delta\Omega\|_F^2, \tag{2}$$

where $\Delta\Omega$ is the Laplacian of Ω. The Laplacian term serves as a regularization to keep the transformation smooth and reduce overfitting. The objective function is convex, so we can solve it and find the global optimal solution. We do a 44-fold cross-validation in order to determine the best regularization coefficient λ. The final average and maximum errors are 39cm and 1.5m in the cross-validation, respectively. For reverse mapping from the WGS84 coordinate to the CAD model, we simply use the Newton-Gaussian algorithm to find the optimal $\mathbf{X}_{\text{CAD}}$ that minimizes $\|\Gamma(\mathbf{X}_{\text{CAD}}, \hat{\Omega}) = \mathbf{X}_{\text{WGS}}\|_2^2$.

Per-Image Fine-Tuning. In the second step, we fine-tune the camera pose for each panorama image. For each image, we first ask the annotator whether this is an indoor image or outdoor image. We discard all of the indoor images. Next, we ask the annotator to label some pairs of corresponding points on the CAD model and the panorama image. We provide a labeling software so that an annotator can switch between the 3D model and the panorama image, click to add a point on them, and optimize the camera pose to minimize the reprojection error. We show the user interface of our annotation tool in Figure 9 of the supplementary material. We instruct the annotator to only put points on roof corners if possible. This is because our CAD model is made from aerial images, and therefore the locations of roof corners are usually much more reliable. We ask annotators to label at least 8 pairs of corresponding points for each viewpoint unless there not exist enough buildings in that scene.

Because we have a good initialization of the camera pose for each panorama image from the IMU data, we apply LevenbergâĂŞMarquardt algorithm to compute the camera pose

[2]Related resources are summarized at https://3d.bk.tudelft.nl/opendata/opencities.

that minimizes the nonlinear angular reprojection error of the corresponding points. Mathematically, let $\mathbf{x}_i \in \mathbb{S}^3$ be the unit vector representing the ray direction of the ith labeled point on the panorama image with respect to the camera and $\mathbf{X}_i \in \mathbb{R}^3$ be the coordinate of the corresponding labeled vertex in the world space of the CAD model. The problem can be formulated as finding the best 6-DoF panorama camera pose Θ (parameterized by its location, azimuth, and up direction) that minimizes the following reprojection error:

$$\min_{\Theta} \sum_{i=1}^{n} \arccos^2 \left(\langle \mathbf{x}_i, \mathbf{P}_{\Theta}(\mathbf{X}_i) \rangle \right), \tag{3}$$

where $\mathbf{P}_{\Theta}$ projects the world-space coordinate to the panorama space $\mathbb{S}^3$ with respect to the camera pose Θ.

5. Experiments

In this section, we will justify the necessity of data platforms based on CAD models, e.g., HoliCity, for 3D vision research. We conduct experiments on the tasks of *surface segmentation* (high-level representation) and *surface normal estimation* (low-level representation) to demonstrate the use of HoliCity and study of its generalizability from and to other datasets. The reason we choose surface segmentation and normal estimation is because previously researchers hardly test their methods on outdoor environments for these tasks. For example, existing works on surface normal estimation [16, 4, 58, 24] only demonstrate their results on indoor scenes. We hypothesize that this is because the quasi-dense and noisy points clouds from outdoor datasets cannot reliably provide the direction of surface normal.

In the task of surface segmentation, an algorithm takes RGB images as input and outputs regions that are considered as a continuous smooth surface, as shown in the second row of Figure 2. Surface segmentation is useful for applications in AR/VR such as object placement. It can be viewed as generalized plane detection [33], in which the results include curved surfaces in addition to flat planes. Prior to HoliCity, methods of plane detection are designed for indoor datasets [33, 32, 69] or synthetic urban scenes [68] only, probably because it is too hard to extract high-quality ground truth planes from noisy point clouds in real-world outdoor datasets (Section 2).

We note that the uses of HoliCity are not limited to the aforementioned tasks. With the existence of CAD models, researchers have the freedom to process and convert our data into a wide range of representations and extract holistic structures. In the supplementary materials, we also report the results of different neural networks on tasks of visual relocalization, 3D plane detection, depth estimation, and vanishing point detection.

5.1. Data Processing

Splitting. We provide two different splits of viewpoints as training, validation, and testing sets.

1. data are split randomly for tasks like relocalization;
2. data are split according to x and y coordinates so that there is no spatial overlap between each set. This is the default split and we use it to study the generalizability of HoliCity on tasks such as normal estimation and surface segmentation.

Rendering. As most existing algorithms take perspective images as input, we provide the perspective renderings for all the viewpoints. For each panorama, we sample 8 views with evenly-spaced (45 degrees apart) yaw angles and randomly sampled pitch angles between 0 and 45 degrees. We use the camera with a 90-degree field of view and render the images with resolution 512×512. We render depth maps, normal maps, and semantic segmentation (Figures 1e and 2) from the CAD model with the same specifications using OpenGL.

Surface Segmentation. One advantage of HoliCity over traditional LiDAR-based outdoor datasets such as KITTI [20] and RobotCar [37] is that the CAD model from HoliCity could provide a structured and accurate representation of surfaces, which makes extracting high-level representations more reliable. Here, we briefly describe our algorithm of extracting the surface segmentation from HoliCity. The sampled results are shown in the second row of Figure 2.

The CAD model in our dataset is represented as a set of polygons of surfaces. We do a breadth-first-search (BFS) from each polygon to compute the surface segments that this polygon belongs to. For each nearby polygon found during the BFS, we add it into the current segments if the (approximated) curvature at the intersection line between the adjacent polygons is less than a threshold value. This threshold controls the minimal curvature required for splitting a surface segment. Because the provided CAD model is not a perfect manifold, we treat two polygons as neighbors if there exists a vertex on each of them whose distance is smaller than a threshold distance. This distance also controls the granularity of the resulting segments. Increasing its value removes small segments.

5.2. Settings and Baselines

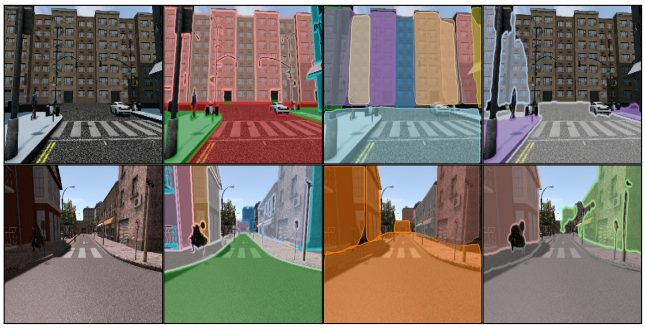
Although it is hard to directly extract high-quality surface segments and normal maps from traditional outdoor datasets, it is still possible to train models on an indoor or synthetic outdoor dataset and then apply them to a real-world outdoor dataset. Therefore, we design experiments to evaluate the feasibility of such an approach and justify the necessity of HoliCity. Besides, we test how well the model trained on HoliCity can generalize to other street-view datasets such as MegaDepth [30].

(a) RGB image of HoliCity (input) (b) Ground truth of HoliCity (segment) (c) [22] trained on HoliCity (segment) (d) [22] trained on ScanNet (segment) (e) [69] trained on HoliCity (segment) (f) [69] trained on ScanNet (segment) (g) [67] trained on SYNTHIA (segment) (h) Ground truth of HoliCity (normal) (i) [46] trained on HoliCity (normal) (j) [46] trained on ScanNet (normal)

Figure 4: Qualitative results of models evaluated on HoliCity. We test models of MaskRCNN [22], Associative Embedding [69], PlaneRecover [67], and UNet [46] that are trained on HoliCity, ScanNet [14], and SYNTHIA [47] on HoliCity.

(a) RGB image of MegaDepth (input) (b) [22] trained on HoliCity (segment) (c) [22] trained on ScanNet (segment) (d) [69] trained on HoliCity (normal) (d) [69] trained on ScanNet (normal) (e) [67] trained on SYNTHIA (segment) (f) [46] trained on HoliCity (normal) (g) [46] trained on ScanNet (normal)

Figure 5: Qualitative results of models evaluated on images from the MegaDepth dataset [30]. We test models of MaskRCNN [22], Associative Embedding [69] and UNet [46] trained on HoliCity, ScanNet, and SYNTHIA. Models are *not* fine-tuned on the targeting dataset (MegaDepth).



(a) RGB image of SYNTHIA (input) (a) Ground truth of SYNTHIA (segment) (b) [22] trained on HoliCity (segment) (c) [67] trained on SYNTHIA (segment)

Figure 6: Qualitative results evaluated on the SYNTHIA dataset. We test our MaskRCNN model and [67] trained on HoliCity and SYNTHIA, respectively.

Datasets. We use HoliCity (ours), ScanNet [14] (indoor), SYNTHIA (synthetic outdoor) as the training datasets. We evaluate the trained models on images from HoliCity, MegaDepth [30], and SYNTHIA. We perform both qualitative and quantitative comparison on HoliCity and SYNTHIA, while we only perform the qualitative comparison on street-view images of MegaDepth because the ground truth surface segmentation and surface normal are not provided.

Surface Segmentation. We include three baseline methods for surface segmentation: MaskRCNN [22], Associative Embedding [69], and PlaneRecover [68]. MaskRCNN is the state-of-the-art method for instance segmentation. We use the implementation from Detectron2 [64] and train the models by ourselves. Associative Embedding is a method for indoor plane detection. We use its official pre-trained model on ScanNet and retrain the Associative Embedding model

Methods	Training Datasets	Surface Segmentation			Normal Est.
		AP_{50}	AP_{75}	mAP	Mean Error
MaskRCNN [22]	HoliCity	42.0	19.8	21.9	
	ScanNet	5.0	0.6	1.7	
Associative Embedding [69]	HoliCity	20.2	8.5	9.9	
	ScanNet	3.3	0.6	1.1	
UNet [46]	HoliCity				22.6°
	ScanNet				46.3°

Table 2: Results of surface segmentation and normal estimation evaluated on the validation split of HoliCity. Methods are trained on HoliCity (our dataset), ScanNet (indoor dataset) [14], and SYNTHIA [47] (synthetic outdoor dataset) and tested on HoliCity *without fine-tuning*. We report the AP metrics for surface segmentation and mean angular error for normal estimation.

on HoliCity from scratches for comparison. PlaneRecover is an approach designed for SYNTHIA [47]. We evaluate its official pre-trained model.

Normal Estimation. We report the performance of UNet [46]. We train the models on all datasets by ourselves.

5.3. Results and Discussions

We show the qualitative results evaluated on the HoliCity dataset of multiple methods in Figure 4, in which we trained the models of MaskRCNN [22], Associative Embedding [69], PlaneRecover [67], and UNet [46] on HoliCity (ours) ScanNet [14] (indoor dataset), and SYNTHIA [47] (synthetic outdoor dataset) on the task of surface segmentation and normal estimation. We find that for both tasks methods trained on ScanNet and SYNTHIA do not generalize well to HoliCity, which is probably due to the domain gap between training sets and testing sets. This can also be verified by the quantitative metrics in Table 2. We can see that the methods trained on indoor or synthetic outdoor datasets perform much worse on real-world outdoor scenes than the methods trained on HoliCity. We conclude that for existing methods such as MaskRCNN and Associative Embedding, a dataset such as HoliCity is necessary for the tasks of surface segmentation and normal estimation in outdoor environments.

We also conduct the cross-dataset experiment on HoliCity and synthetic SYNTHIA datasets for surface segmentation. In this experiment, we use the official plane detection model trained on SYNTHIA from [67] and train the MaskRCNN [22] model on HoliCity. Then, we evaluate both models on HoliCity and SYNTHIA. We show the quantitative results in Table 3. We find that the model trained on HoliCity can generalize to a synthetic outdoor dataset such as SYNTHIA well, while the model trained on SYNTHIA completely fails on HoliCity. Such observations also apply to the qualitative results in Figures 4 and 6, where the HoliCity-trained model recovers most of the building

Training Datasets (Methods)	Testing Datasets (AP_{50})	
	HoliCity	SYNTHIA
HoliCity (MaskRCNN [22])	42.0	36.1
SYNTHIA (PlaneRecover [67])	1.90	40.6

Table 3: Results of surface segmentation cross-trained and evaluated on the validation split of HoliCity and SYNTHIA [47]. We test our MaskRCNN model [22] trained on HoliCity and the official PlaneRecover model trained on SYNTHIA from [67]. Models are *not fine-tuned* on testing datasets.

surfaces in SYNTHIA despite the differences between the definitions of surface segments in HoliCity and planes in [68]. We hypothesize that the causes of these phenomena are due to the wider variety of scenes covered by HoliCity, compared to the scenes from SYNTHIA.

In fact, not only methods trained on ScanNet and SYNTHIA does not generalize well to HoliCity, they generalize well to images from other outdoor datasets as well, such as MegaDepth [30], as shown in Figure 5. In comparison, methods trained on HoliCity can produce much better surface segmentation and normal maps on these images, which shows HoliCity's potential generalizability to general outdoor imagery.

Finally, we summarize our observations as followings:

1. previous research of plane detection and normal estimation hardly experiments on outdoor datasets;
2. HoliCity can provide both high-quality holistic structures (e.g., surface segments) and low-level representations (e.g., normal maps) of urban environments;
3. models trained on indoor or synthetic outdoor datasets cannot generalize well to real-world outdoor datasets;
4. models trained on HoliCity can generalize to synthetic outdoor scenes;
5. models trained on HoliCity can generalize to real-world street-view imagery from a different dataset.

These observations indicate that HoliCity is an indispensable data platform of urban environments for future research of 3D vision.

6. Conclusions and Future Work

In this work, we introduce a novel city-scale dataset HoliCity that consists of highly accurate annotation between a large set of 2D panorama images and the associated 3D CAD models. The established rigorous annotation pipeline and tools developed may allow us to continue to increase the scale and richness of the dataset in the future. This dataset can support studying and evaluating a wide spectrum of 3D vision methods, from low-level to high-level. Our careful evaluation of the state of art methods (trained on existing and our datasets) has revealed serious lack of generalizabil-

ity of existing methods, especially to a large-scale diverse outdoor dataset like ours. Arguably the greatest value of this dataset is to support developing new methods than can learn to exploit holistic or semantic structures of the scene to achieve highly accurate and robust reconstruction, localization, and augmentation within a city-scale environment. Currently, HoliCity focuses on modeling the building architectures. In the future, we will also extend the datasets to model the moving objects in urban environments, including pedestrians and automobiles.

A. Supplementary Material

A.1. Random Sampled Visualization of HoliCity

In order to visualize the overall alignment quality of our dataset, we show **random sampled** images from HoliCity overlaid with the surface segmentation in Figure 7 (high-resolution CAD models) and Figure 8 (low-resolution CAD models). For the overlays from low-resolution CAD models, there are slightly more model errors and less details compared to the overlays from high-resolution CAD models, especially for the regions near the ground.

A.2. Labeling Tools

As introduced in Section 4, we build the correspondence between the CAD models and the panorama images through labeling pairs of corresponding points on them. Figure 9 shows our labeling tool. The annotators use our labeling tool to put points on the images and models. Within our labeling tool, they can freely navigate in the London city with keyboard shortcuts and adjust the camera pose on the left control panel. They can switch between 3D models and re-projected panorama images with keyboard shortcuts. To add points to the CAD models, users could click around the vertices of the mesh and the annotation tool will automatically snap the point to the nearest visible vertices. The users could also add points on the panorama images with mouse clicks. When the annotator thinks he has labeled enough points, he could optimize the camera pose with current correspondence by clicking the "Optimize" button on the left control panel.

We instruct annotators to label points only on the corners of the building roof if possible. This is because the CAD models are made from aerial images, and the roof features of the CAD model are usually much more reliable. For current batches, each image contains at least 8 pairs of labeled corresponding pairs unless buildings in the panorama images are highly occluded.

A.3. Monocular Depth Estimation

Monocular depth estimation has been a popular task since the beginning of deep learning [16]. To demonstrate our work can better support this task than synthetic datasets, we run the following benchmarks: We train UNet [46] on HoliCity and SYNTHIA (synthetic outdoor) and test them on the validation splits of HoliCity, SYNTHIA, and scene 162 of MegaDepth. We show the qualitative and quantitative results in Figure 10 and Table 4. The goal is to compare the generalizability of HoliCity-trained models and the SYNTHIA-train models. We report the scale-invariant error (SIL) [16] because depth maps of MegaDepth only have relative scales due to the usage of SfM. Here we have two observations. First, methods evaluated on HoliCity has larger errors than the methods evaluated on SYNTHIA. This might be because scenes of HoliCity has more varieties than the scenes of SYNTHIA. Second, methods trained on HoliCity has better performance when tested on other outdoor datasets (MegaDepth) than methods trained on SYNTHIA. This shows that HoliCity-trained models have better generalizability than that of SYNTHIA-trained models, which is probably because images in HoliCity are more realistic and versatile than the images from the synthetic dataset.

Training Datasets	Testing Datasets (SIL [16] Error)		
	HoliCity	SYNTHIA	MegaDepth
HoliCity	0.101	0.237	0.088
SYNTHIA	0.353	0.054	0.246

Table 4: Results of monocular depth estimation cross-trained and evaluated on the validation splits of HoliCity, SYNTHIA, and scene 162 of MegaDepth.

A.4. Vanishing Point Detection

Vanishing points are an important concept in 3D vision, especially in outdoor urban environments. This is because vanishing points provide information about camera poses with respect to local building structures. Generating vanishing points with CAD models is relatively easy and accurate. First, because all the scene contains the upward vertical vanishing points, we compute them by projecting the up vector $(0, 0, 1)$ into the image with its camera pose. Second, we need to find horizontal vanishing points. We solve this by clustering the surface normal with DBSCAN [7] and find the direction of horizontal vanishing points by computing the cross product of the surface normal and the up vector. We use DBSCAN because it does not require a predetermined number of clustering centers, and we want the number of horizontals vanishing variable. This avoids the limitation of the strong Manhattan assumption that is used by previous datasets including YUD [15], ECD [5], and HLW [63].

We test two algorithms, the conventional LSD [56, 19] + J-Linkage [54] and the recent learning-based NeurVPS [71]. The former uses line segment detectors to detect the lines and then clusters them according to their intersection using

Figure 7: **Random sampled** perspective images overlaid with surface segments from **high-resolution** CAD models.

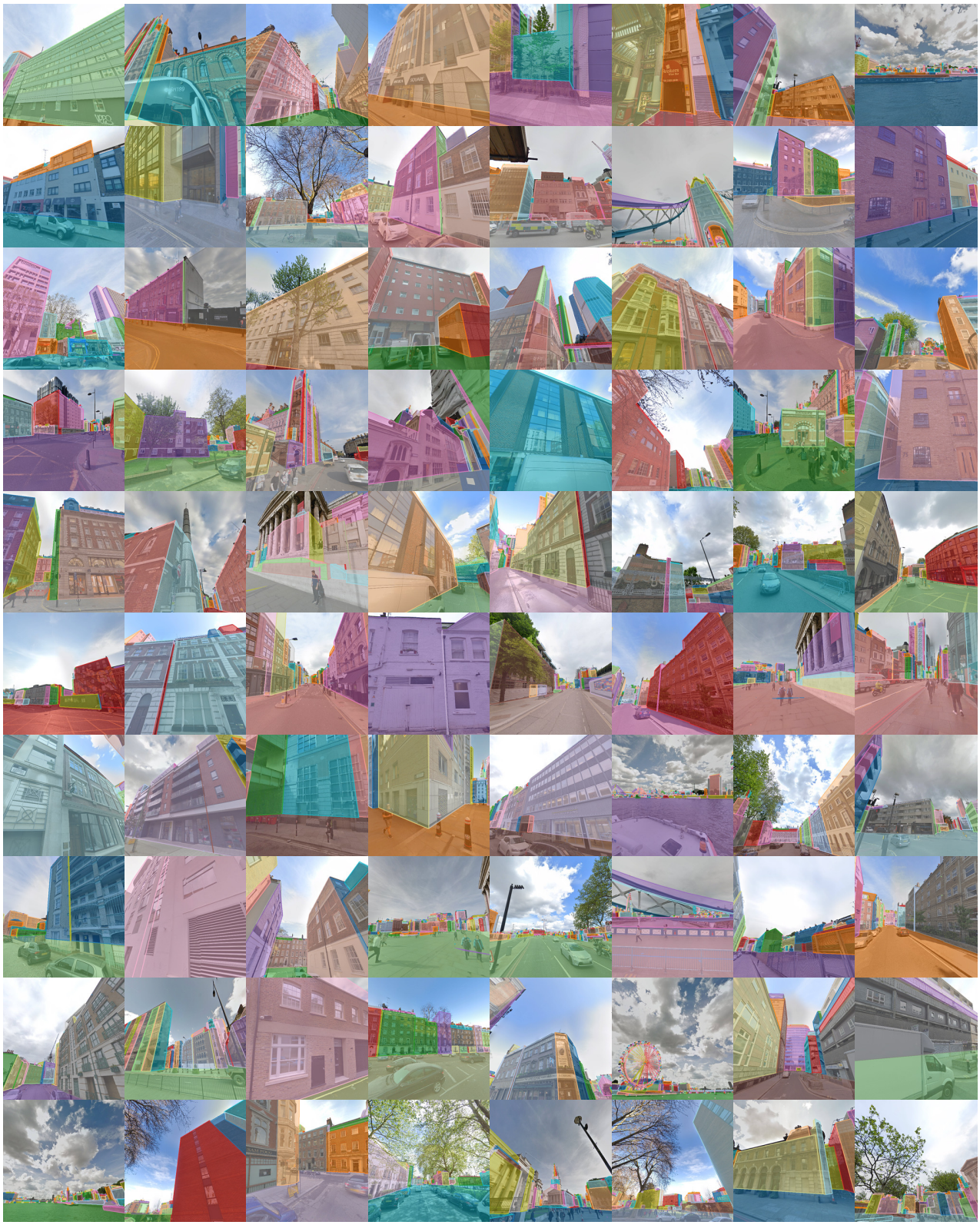
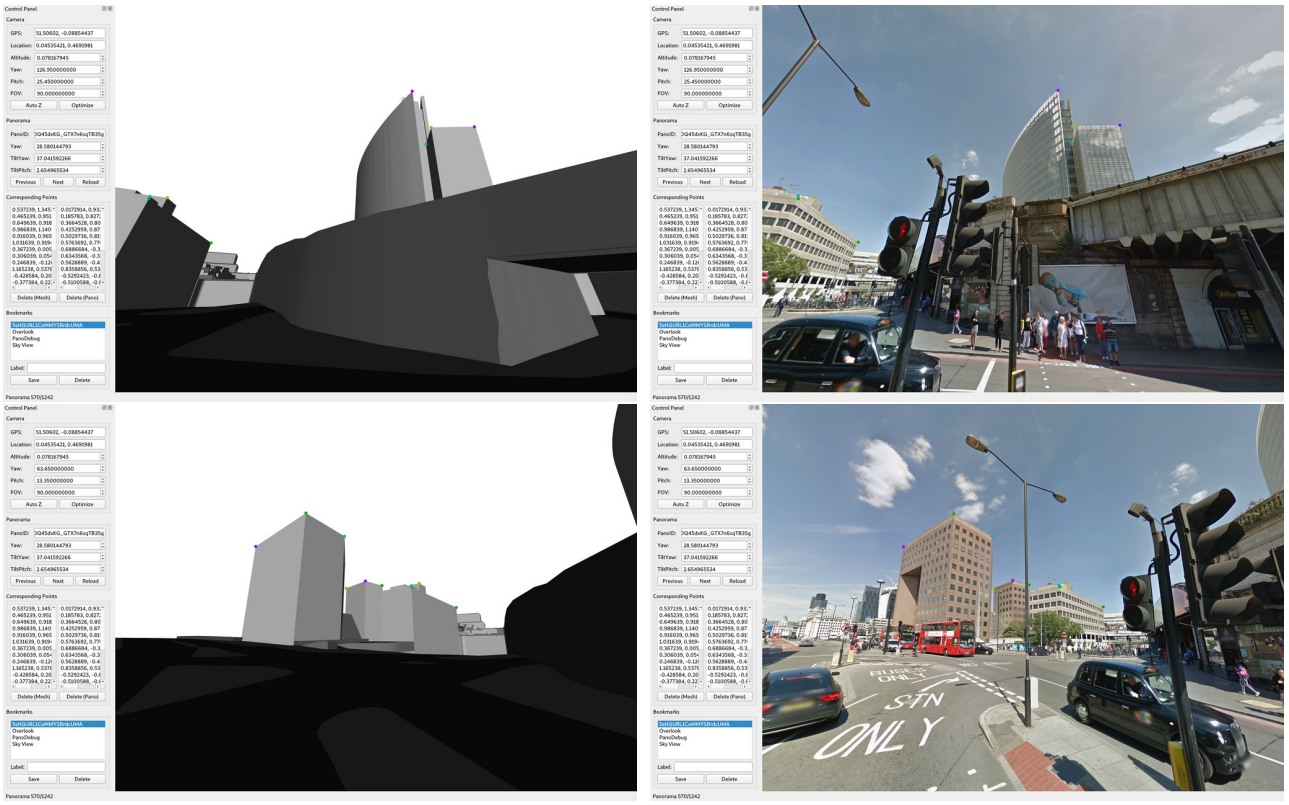
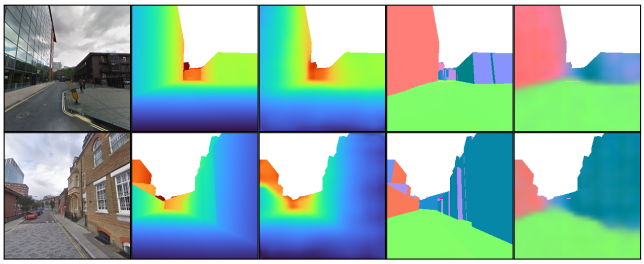


Figure 8: **Random sampled** perspective images overlaid with surface segments from **low-resolution** CAD models.



(a) UI when annotating the 3D model. (b) UI when annotating panorama images.

Figure 9: User interface for panorama-to-model corresponding point annotation.



(a) RGB image (input) (b) Ground truth (depth) (c) Inferred [46] (depth) (d) Ground truth (normal) (e) Inferred [46] (normal)

Figure 10: Visualization of results on tasks of monocular normal and depth estimation. Models are trained and evaluated on HoliCity.

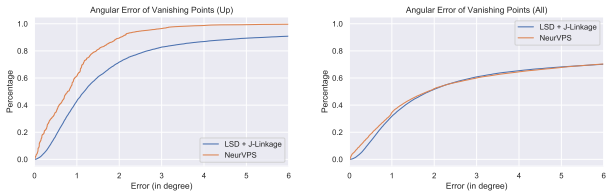


Figure 11: Angular errors of vanishing points. The left figure shows the plot of angular errors vs. percentages of algorithms when predicting the upward vanishing points, while the right figure shows the plot of angular errors vs. percentages of algorithms when predicting all the vanishing points.

J-Linkage. NeurVPS, on the other hand, uses a coarse-to-fine strategy and tests whether a vanishing point candidate is valid with a conic convolutional neural network. Figure 11 shows the results of both algorithms. The median error of NeurVPS is around 0.5 degrees for the up vanishing points and 1.5 degrees for all vanishing points. This shows that the camera pose of our dataset should at least be around that accuracy. Besides, we find that learning-based NeurVPS has the similar performance as LSD when considering all the vanishing points. This is probably because NeurVPS has not yet been optimized for detecting a variable number of vanishing points.

A.5. Relocalization

Precise camera localization from images is key to many 3D vision tasks, such as visual compasses, navigation, au-

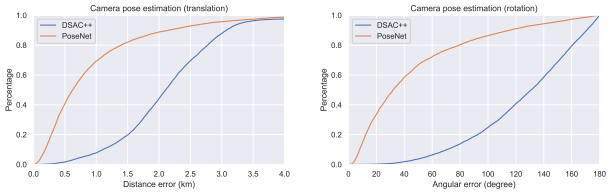


Figure 12: Orientation and Localization errors for existing camera localization methods trained and tested using HoliCity dataset.

tonomous driving, and augmented reality. Due to the cost, existing outdoor datasets only capture limited regions [49] or a few paths [36]. In comparison, HoliCity provides an accurate camera pose densely sampled in an entire city with various view angles. We believe that a method with decent performance on our dataset will be useful for many real-world applications. Here, we benchmark state-of-the-art methods on our dataset and find a significant gap between the existing techniques and real-world challenges.

We choose to evaluate two recent deep learning-based methods, namely PoseNet [27] and DSAC++ [8] as two representative works related to direct regression and scene coordinate estimation. Figure 12 summarizes the errors in the camera distance and orientation predicted. The Y-axis represents the threshold of the prediction error and the X-axis is the percentage of frames in the test data with prediction error smaller than the threshold. The mean errors in location for PoseNet and DSAC++ are 921m and $2,086$m, respectively. The mean orientation errors for them are 47.1° and 124.2°, respectively. Although such results look problematic, we do try our best to tune the parameters of the networks and find it hard to reach a reasonable performance on HoliCity. We do observe that in the original paper, their results on scenes such as "Office" have similar accuracy as ours. We think that the difficulty of HoliCity for relocalization comes from its massive scale (20km^2), relatively large baseline (Figure 1b), and long span (Section 3), hence resulting in huge prediction errors for these learning-based relocalization approaches.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv*, 2017.

[2] Minoru Asada, Masahiko Yachida, and Saburo Tsuji. Analysis of three-dimensional motions in blocks world. *Pattern Recognition*, 17(1):57–71, 1984.

[3] Ruzena Bajcsy and Lawrence Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5(1):52–67, 1976.

[4] Aayush Bansal, Bryan Russell, and Abhinav 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.

[5] Olga Barinova, Victor Lempitsky, Elena Tretiak, and Pushmeet Kohli. Geometric image parsing in man-made environments. In *ECCV*, 2010.

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

[7] Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[8] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018.

[9] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv*, 2015.

[11] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[12] Maxwell B Clowes. On seeing things. *Artificial intelligence*, 2(1):79–116, 1971.

[13] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niebner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

[15] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008.

[16] 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.

[17] Gilbert Falk. Interpretation of imperfect line data as a three-dimensional scene. *Artificial intelligence*, 3:101–144, 1972.

[18] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[19] Chen Feng, Fei Deng, and Vineet R Kamat. Semi-automatic 3D reconstruction of piecewise planar building models from single image. *CONVR*, 2010.

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 2013.

[21] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. AtlasNet: A papier-mache approach to learning 3D surface generation. *arXiv preprint arXiv:1802.05384*, 2018.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[23] Jingwei Huang, Angela Dai, Leonidas J Guibas, and Matthias Nießner. 3Dlite: towards commodity 3d scanning for content creation. *ACM Trans. Graph.*, 36(6):203–1, 2017.

[24] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas Guibas. FrameNet: Learning local canonical frames of 3D surfaces from a single RGB image. In *ICCV*, 2019.

[25] Takeo Kanade. A theory of origami world. *Artificial intelligence*, 13(3):279–311, 1980.

[26] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[27] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017.

[28] Thomas H Kolbe, Gerhard Gröger, and Lutz Plümer. Citygml: Interoperable access to 3d city models. In *Geo-information for disaster management*, pages 883–899. Springer, 2005.

[29] Shih Jong Lee, Robert M Haralick, and Ming Chua Zhang. Understanding objects with curved surfaces from a single perspective view of boundaries. *Artificial Intelligence*, 26(2):145–169, 1985.

[30] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[32] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[33] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka 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.

[34] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[35] AK Macworth. Interpreting pictures of polyhedral scenes. *Artificial intelligence*, 4(2):121–137, 1973.

[36] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[37] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[38] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. the MIT press, 1982.

[39] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. SceneNet RGB-D: Can 5M synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, 2017.

[40] A. Mousavian, A.toshev, M. Fiser, J. Kosecka, A. Wahid, and J. Davidson. Visual representations for semantic target driven navigation. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[41] Raúl Mur-Artal, JMM Montiel, and Juan D Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 2015.

[42] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*, 2019.

[44] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[45] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[47] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[48] Ruth Shapira. More about polyhedra-interpretation through constructions in the image plane. *IEEE transactions on pattern analysis and machine intelligence*, PAMI-7(1):1–16, 1985.

[49] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[50] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[51] Kent A Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17(1-3):47–73, 1981.

[52] Kokichi Sugihara. Picture language for skeletal polyhedra. *Computer Graphics and Image Processing*, 8(3):382–405, 1978.

[53] Kokichi Sugihara. Mathematical structures of line drawings of polyhedronsâĂŤtoward man-machine communication by means of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(5):458–469, 1982.

[54] Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with J-linkage. In *European conference on computer vision*, pages 537–547. Springer, 2008.

[55] Matthias Uden and Alexander Zipf. Open building models: Towards a platform for crowdsourcing virtual 3d cities. In *Progress and new trends in 3D geoinformation sciences*, pages 299–314. Springer, 2013.

[56] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *PAMI*, 2010.

[57] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019.

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

[59] Walter Whiteley. Realizability of polyhedra. *Structural topology, 1979, núm. 1*, 1979.

[60] Walter Whiteley. Motions and stresses of projected polyhedra. *Structural Topology 1982, núm 7*, 1982.

[61] Andrew P Witkin and Jay M Tenenbaum. On the role of structure in vision. In *Human and machine vision*, pages 481–543. Elsevier, 1983.

[62] Robert J Woodham. Analysing images of curved surfaces. *Artificial Intelligence*, 17(1-3):117–140, 1981.

[63] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. In *BMVC*, 2016.

[64] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[65] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013.

[66] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[67] Fengting Yang and Zihan Zhou. Recovering 3D planes from a single image via convolutional neural networks. In *ECCV*, 2018.

[68] Fengting Yang and Zihan Zhou. Recovering 3D planes from a single image via convolutional neural networks. In *ECCV*, 2018.

[69] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1029–1037, 2019.

[70] Huayi Zeng, Kevin Joseph, Adam Vest, and Yasutaka Furukawa. Bundle pooling for polygonal architecture segmentation problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 750–759, 2020.

[71] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. NeurVPS: Neural vanishing point scanning via conic convolution. In *Advances in Neural Information Processing Systems*, pages 864–873, 2019.

[72] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *ICCV*, 2019.

[73] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3D Manhattan wireframes from a single image. *ICCV*, 2019.

[74] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. LayoutNet: Reconstructing the 3D room layout from a single RGB image. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, June 2018.