# Single View Scene Scale Estimation using Scale Field

Byeong-Uk Lee[1] Jianming Zhang[2] Yannick Hold-Geoffroy[2] In So Kweon[1]
[1]KAIST, Republic of Korea [2]Adobe Inc., USA

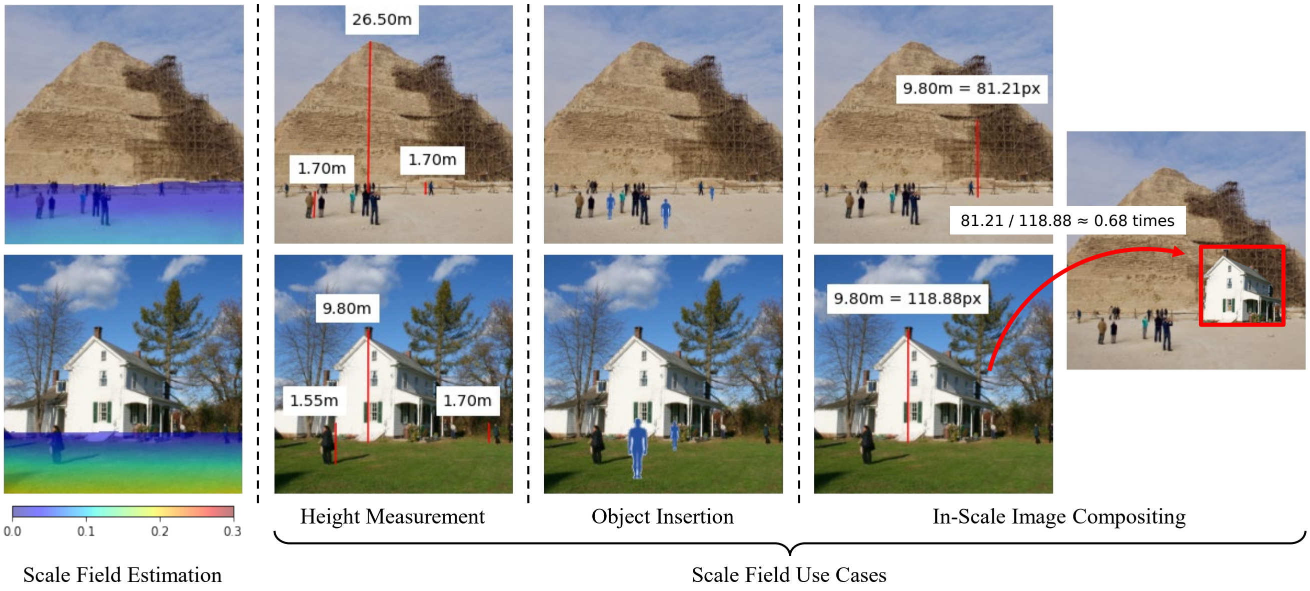


Figure 1. Scale field estimation results and their use cases.

## Abstract

*In this paper, we propose a single image scale estimation method based on a novel scale field representation. A scale field defines the local pixel-to-metric conversion ratio along the gravity direction on all the ground pixels. This representation resolves the ambiguity in camera parameters, allowing us to use a simple yet effective way to collect scale annotations on arbitrary images from human annotators. By training our model on calibrated panoramic image data and the in-the-wild human annotated data, our single image scene scale estimation network generates robust scale field on a variety of image, which can be utilized in various 3D understanding and scale-aware image editing applications.*

## 1. Introduction

Single image 3D understanding plays a significant role in various computer vision tasks, such as AR/VR applications, robotics, and computational 3D photography. Many recent approaches show promising results in estimating depth [17, 18, 25], scene structure [24, 32], or radiance field [20, 22, 23]. However, they often treat scale as an undetermined factor as it is a very ill-posed problem when the physical extrinsics of the camera is unknown. Therefore, it remains a very challenging task to estimate the metric scale of a scene for a single unconstrained image.

One seminal work in scale estimation is single view metrology [6]. As detailed in this line of works [6, 10, 35], knowing horizon line, field of view (FoV) and absolute camera height enables the conversion between any 2D measurements in image space to 3D measurements. Horizon line and FoV can be estimated using visual features, as many of other previous methods suggest [9, 11, 29, 30, 35]. However, since the absolute camera height information cannot be obtained from low-level visual features, Criminisi *et al.* [6], Hoiem *et al.* [10] and Zhu *et al.* [35] utilize canonical object heights as reference. Reprojecting the 2D bounding boxes of well-known objects like humans or cars to 3D space with their known metric heights, these methods derive the vertical height of the camera, and then calculate metric scale of other objects in the 2D image.

However, there are two major downsides of single view metrology based approaches. First, their scale estimation relies on the existence of known objects with either precise or canonical metric heights. This restricts practical usage of their methods because there may or may not be an object with known height. In addition, the height of a known object can also change with its pose (*e.g.* a sitting person *vs* a standing person), making the system less reliable. Second, their prediction of the scene scale is highly dependent on a few global parameters like horizon line, FoV, and camera height. Among these parameters, camera height is very difficult to directly predict based on visual features. Moreover, models that directly predict these parameters tend to overfit to a certain dataset where the camera height and the FoV lie in a limited range.

In this paper, to tackle the aforementioned issues, we introduce a novel way of representing scene scale. We define a pixel-wise pixel-to-metric 2D map, called **Scale Field**. Using this local and dense representation, our goal is to train a robust and generalizable scale estimation network to recover the scene scale. Also, we provide an efficient pipeline to annotate scale fields on random images based on our geometrical observations consistent with single view metrology. This allows us to collect a diverse set of images with varying camera angles and camera heights. Our single-view scene scale estimation network trained on this dataset shows robust results on a variety of scenes, and it generalizes well even on object-centric images. With our predicted scale field, we can perform various 3D scene understanding and scale-aware image editing, as visualized in Fig. 1.

To summarize, our major contributions are:

- We introduce **Scale Field**, a novel representation of scene scale information, which can be utilized in various 3D understanding and scale-aware image editing tasks.
- We propose a pipeline to annotate scale fields on web images, and collect a diverse set of training samples.
- We provide a formulation of the **single image scene scale estimation network**, trained on a mixture of training data from panoramic images and human annotated data. Our method shows great robustness and generalizability on in-the-wild images.

## 2. Related Work

**Scale Estimation.** Obtaining scene scale can be done by estimating scale-aware information. The most straightforward approach is to predict pixel-wise depth map from the scene. Monocular depth estimation [3, 16, 17] has improved by a large margin alongside technical improvements on powerful network backbones and deep learning training schemes. Still, some of them utilize additional information like camera parameters and surface normal to maintain consistency between 2D depth map and 3D structure. Moreover, they are still far behind in terms of accuracy compared to depth completion [14, 19], stereo matching [5, 21] and multi-view stereo [13, 18, 27, 31, 34], which exploits more direct scale-aware priors like sparse depth measurements or multi-view constraints.

**Camera Calibration.** Camera intrinsic and extrinsic parameters enable conversion between 2D measurements and 3D measurements. Intrinsic camera parameter estimation methods [9, 11, 29] are usually done by using an explicit geometric model of a camera, or data-dependant supervised training scheme. Extrinsic camera parameters consist of rotation and translation. Rotation such as camera pitch or roll can be easily interpreted as horizon line, which can be retrieved in a classical way of utilizing line segments and vanishing points [2, 8, 15, 33], or in deep learning-based way by learning from large-scale dataset with ground truth horizons [11, 30]. However, predicting absolute translation, which is crucial in recovering scale, is difficult without using additional annotations.

**Single View Metrology.** Single view metrology [6] establishes the relationship among low-level image features like vanishing points and vanishing lines, 3D sizes of objects in the 3D space, and projected 2D positions and sizes. Recent works [10, 12, 35] that follow this philosophy estimate either unknown heights of objects in the scene or camera height by using camera parameters, image features, and at least one 2D and 3D size annotation of reference objects.

## 3. Introducing Scale Field

### 3.1. Geometric Background

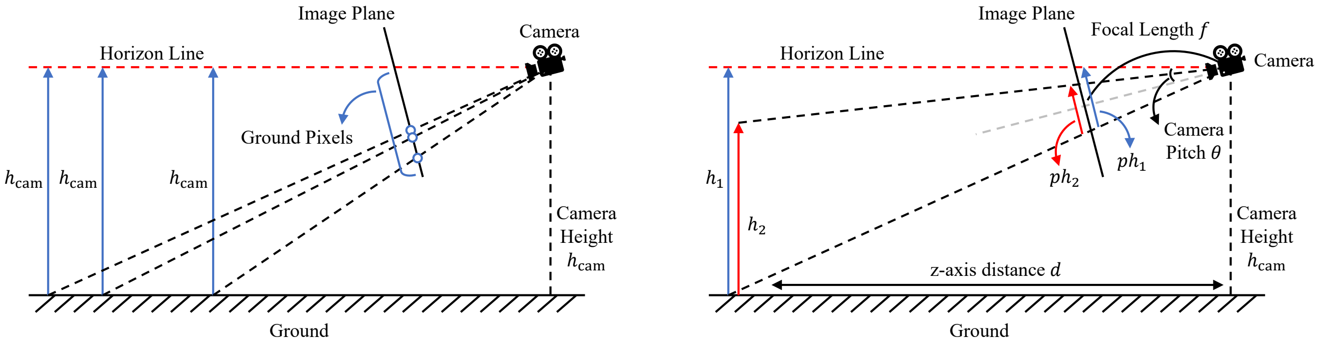
Before describing our scale field, we first review the geometric background regarding the camera and metric scale. Our observations are summarized in two-fold.

- All the vertical lines from ground pixel to horizon line are reprojected to 3D space with same metric heights, which is also same as camera height.
- In a fixed ground position, metric height and pixel height are approximately in a linear relation.

Fig. 2 visually describes the relationship between the camera, ground, and horizon line.

**Observation 1.** Assuming 'ground' as a single plane where the camera height was measured, in 2D image, we denote all the pixels under the horizon line as ground pixels. We focus on the vertical lines, *i.e.*, aligned with gravity direction, from ground pixel to horizon line, as in Fig. 3c. When they are reprojected to 3D space, all of these lines will have same metric heights. And that height is also same as the height of the camera, $h_{\text{cam}}$, as shown in Fig. 2a.

**Observation 2.** We additionally inspect the relationship between pixel height and metric height on a fixed ground point. As illustrated in Fig. 2b, we denote pitch and focal length of the camera as $\theta$ and $f$ respectively, and the z-axis



(a)

(b)

Figure 2. **Visualization of the relationship between pixel height, metric height, and camera height.** (a) All the vertical lines from the ground to the horizon line will have the same metric height as camera height. (b) From the same ground position, metric height and pixel height are approximately in linear relation.

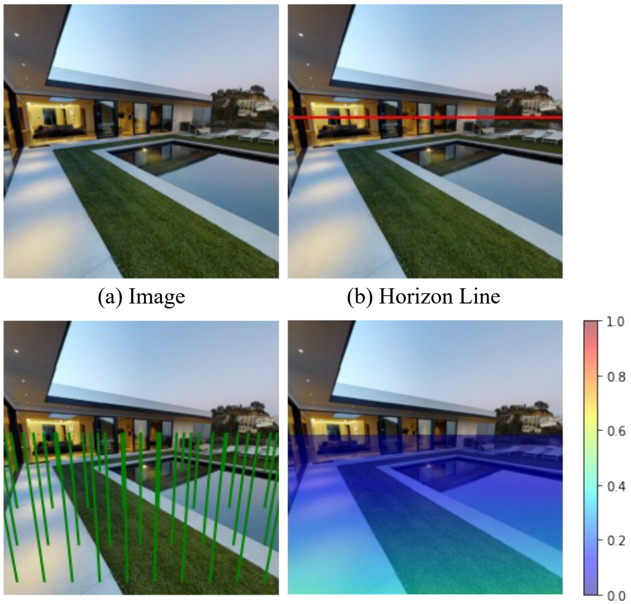
distance from the camera to the ground point as $d$. Blue arrow with metric height $h_1$ and the red arrow with metric height $h_2$ are set to start from the same ground point. Pixel heights of the projected arrows in 2D image are defined as $ph_1$ and $ph_2$, respectively. Then, we can express $ph_1$ and $ph_2$ in terms of $h_1$ and $h_2$ like following:

$$\begin{aligned}
ph_1 &= h_1 \frac{f \cos\phi_1}{d\cos\theta\cos(\phi_1-\theta)}, \\
ph_2 &= h_2 \frac{f\cos\phi_1(\cos\theta+\sin\theta\tan\phi_2)}{d\cos(\phi_1-\theta)}, \\
\phi_1 &= \arctan\frac{h_1}{d}, \quad \phi_2 = \theta - \arctan\frac{h_1-h_2}{d}.
\end{aligned} \tag{1}$$

Following approximations applied in [10], the metric height to pixel height formulation can be derived from Eq. (1), as in,

$$\frac{ph_1}{ph_2} \approx \frac{h_1}{h_2}. \tag{2}$$

This means that the metric height and the pixel height from a single ground pixel are in linear relation.



(a) Image

(b) Horizon Line

(c) Ground2Horizon

(d) Scale Field

Figure 3. **Components for the scale field generation.** Overlaid colormap of scale field in (d) are represented relatively to image height. For example, red color with scale field value of 1.0 refers to image height amount of pixels equal to 1.0m. Note that scale field values converge to 0.0 near the horizon line.

## 3.2. Scale Field Formulation

Based on the above observations, we now formulate our scale field. We first calculate a 2D vector field in an image where each vector starts from the ground pixel and ends at the intersection to the horizon line, which we call **ground2horizon**. Each ground2horizon vector should be perpendicular to the ground plane when reprojected to 3D space. As was mentioned in Sec. 3.1, all the ground2horizon vectors obtained in a single image have the same metric height, and each of the vectors will have linear relation between pixel and metric heights. We simply define the **Scale Field** $SF$ by dividing the pixel magnitudes of these vectors by the absolute metric height of the camera, as in,

$$SF(x,y) = \begin{cases} ph(x,y)/h_{\text{cam}} & \text{if } (x,y) \in \text{ground}, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $(x,y)$ is a 2D coordinate and $ph$ is a pixel height of the ground2horizon vector from $(x,y)$, normalized by image height and width. This makes our scale field a 2D map of per-pixel pixel-to-metric ratio.

For each ground pixel in an image, the scale field provides information on how many pixels represent a certain amount of vertical metric length. It enables various scale-coherent applications on 2D images such as 3D scene understanding or scale-aware image editing.

## 4. Single Image Scene Scale Estimation

In this section, we introduce our single image scene scale estimation network and its training pipeline using scale field. We first describe the dataset generation scheme using panoramic images and our method of collecting web image annotations in Sec. 4.1.

### 4.1. Dataset Generation

In most neural network-based algorithms, it is very crucial to acquire an appropriate training dataset to give the neural network reliable supervision. In order to generate a scale field, metric camera height, horizon line, and ground2horizon vectors should be prepared. In addition, to make our scene scale estimation network robust to a wide variety of images, training samples with various ranges of aforementioned parameters should be used.

All of the values can be obtained from intrinsic and extrinsic camera parameters. However, many of the public vision datasets with both intrinsic and extrinsic camera parameters are mostly taken handheld or while attached to the grounded vehicle. Moreover, one dataset is most likely captured using the same sensor system. Therefore, utilizing these datasets will lack variety in terms of FoV, roll, pitch, and especially, camera height.

**Panorama Images.** Due to the problems of using public datasets of perspective images, we choose to utilize 360° panoramic images. The most beneficial property of panorama is its scalability. When performing equirectangular-to-perspective cropping, it is able to make a large number of perspective images from a single panorama by using different combinations of camera parameters, while maintaining the same camera height.

We select the Stanford2D3D [1] and Matterport3D [4] datasets for our scale field dataset generation. They are two of the most frequently used panorama datasets with camera height provided. For each panorama image, we sample 40 different perspective images and scale field pairs. We randomly select camera parameters from uniform distributions of $[45°, 105°]$ for FoV, $[-30°, 30°]$ for roll, $[-15°, 15°]$ for elevation and $[-180°, 180°]$ for azimuth. We use the official train/test splits from each of the datasets, and randomly select only 2000 pairs for testing from each dataset. Fig. 4 shows examples of different perspective crops and their horizone line, ground2horizon vectors and scale field from a single panorama image.

However, both Stanford2D3D and Matterport3D datasets only contain indoor scenes. To provide more visual variety to the training pool, we also utilize additional indoor/outdoor panorama images of our own, applying the same process as formerly described.

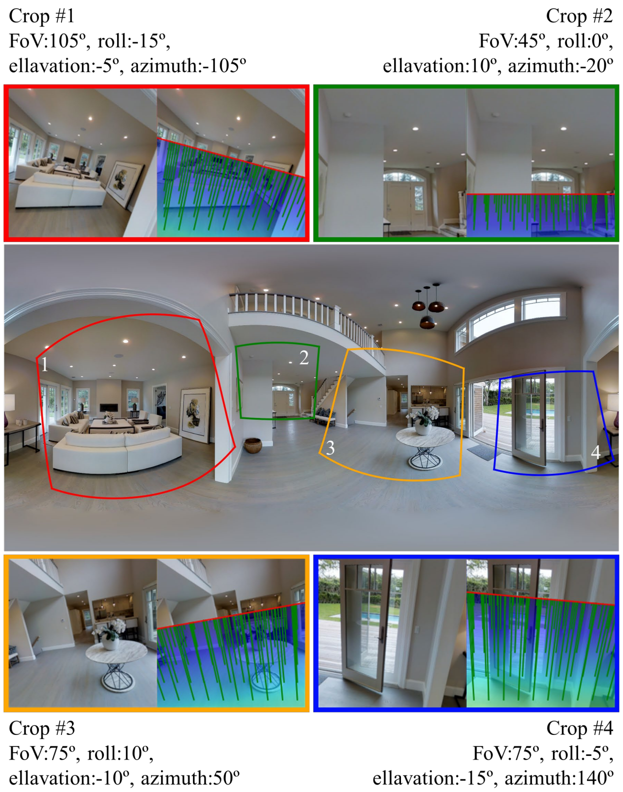

Crop #1
FoV:105º, roll:-15º,
ellavation:-5º, azimuth:-105º
Crop #2
FoV:45º, roll:0º,
ellavation:10º, azimuth:-20º
1
2
3
4
Crop #3
FoV:75º, roll:10º,
ellavation:-10º, azimuth:50º
Crop #4
FoV:75º, roll:-5º,
ellavation:-15º, azimuth:140º

Figure 4. Examples of different perspective crops using different sets of camera parameters from a single panorama image.

**Web Images.** While panorama-based datasets provide a sufficient amount of training samples, there exists a major problem that may hinder the generalizability of the network. Most of the panoramic datasets are taken from 360° cameras that are placed on the tripod. This means that many of the training/testing samples have similar camera heights. For example, the mean and the standard deviation values of camera heights from all the samples from panorama images are 1.538m and 0.069, respectively. Since we want our scene scale estimation network to perform well on arbitrary scenes, we need an additional source for our scale field dataset.

In order to alleviate this issue, we introduce another method for collecting the scale field annotations on random web images. Fig. 5 describes the overall pipeline of the annotation process. We first annotate the horizon line. This can be done by using at least a pair of horizontal parallel lines in the image and identifying their vanishing point. We assume the image to have nearly-zero roll or otherwise rectified beforehand.

The next step is to annotate ground-attached vertical lines with known or estimated metric heights. In Fig. 5,

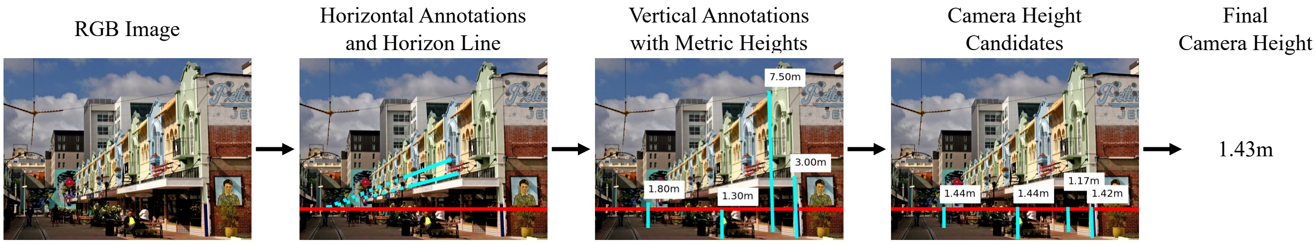


Figure 5. Scale field annotation pipeline using web images.

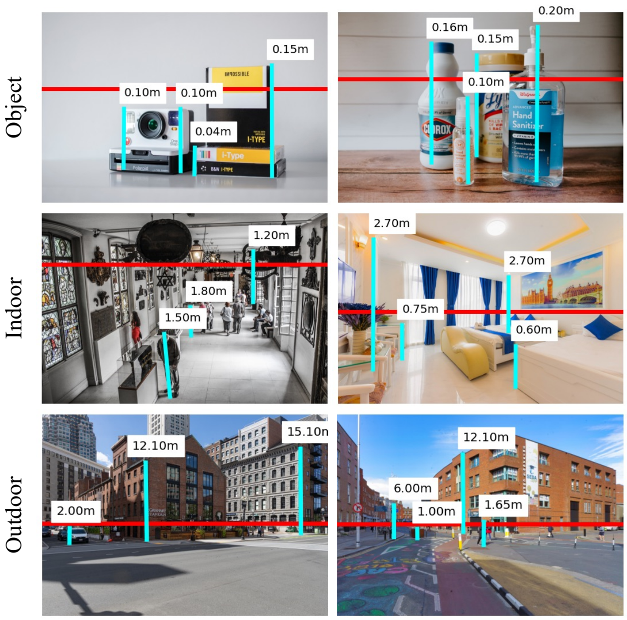


Figure 6. Examples of annotated web images.

for example, the height of the human and the building are annotated to have 1.8m and 7.5m heights. Using these vertical lines, horizon line, and the linearity between metric and pixel heights described in Eq. (2), we can obtain camera height candidates. The final camera height is the median value of all the candidates from the annotations. With the annotations and camera height, all the components for our scale field are calculated using formulations previously described in Sec. 3.1 and [9, 11, 29].

We annotated three categories of web images, which are object, indoor and outdoor. The 'object' category contains object-centric images such as close-up images or tabletop images. Our web image dataset contains a total of 1372 images, with 120 objects, 981 indoor, and 271 outdoor images. Fig. 6 shows some examples of our web image dataset and their annotations.

Obtained camera heights of these images range from 6cm to 234m. We handpicked 274 images from three categories for evaluation, and utilized others for training. For both training and testing, we square-cropped the image to three crops, so that every crops can cover the whole image. Note that we do not utilize image crops with horizon line lying below image bottom boundary, since we cannot define scale field in this circumstance.

## 4.2. Network Architecture

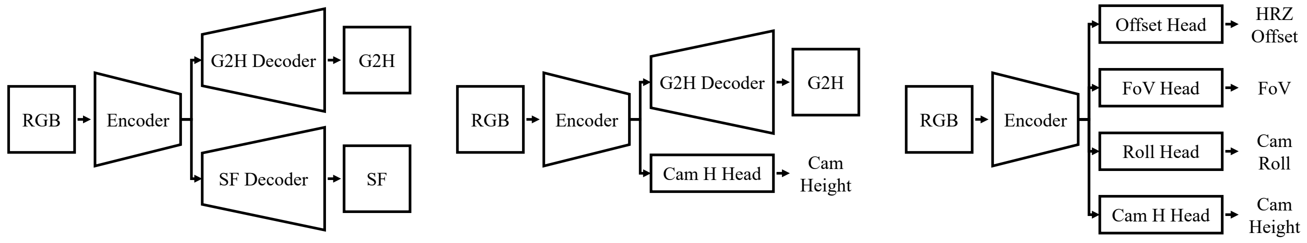
We now describe our single-view scene scale estimation network using scale field. We design our network to simultaneously predict both ground2horizon and scale field, so that it can be aware of not only scale but also the gravity direction. As was introduced in Sec. 3.2, ground2horizon is also a local and dense representation that implicitly holds information about the camera parameters.

To show the robustness of predicting local fields instead of global parameters, our scene scale estimation network has 3 variants: 1) ground2horizon and scale field estimation (G2H+SF, Fig. 7a), 2) ground2horizon and camera height estimation (G2H+CamH, Fig. 7b), and 3) horizon line, field of view and camera height estimation (CamParams, Fig. 7c). Variant CamParams is our own implementation version of single image camera calibration method by Hold-Geoffroy *et al.* [11] with the addition of the camera height estimation head. Model G2H+SF only predicts local fields, while model CamParams only predicts global parameters and G2H+CamH is the mixture of both. All three models, using the outputs of their own, are able to construct ground2horizon and the scale field, which will be used in quantitative evaluations. Overall structures of our model and its variants are illustrated in Fig. 7.

All the variants have the same network architecture of PVTv2 [28], which is a transformer-based feature extractor. Decoder architecture for 2D field estimation follows the design of UNet [26] that outputs the same resolution as the input image, while the number of channels differs according to the target output value. The estimation head for the global parameter is a fully connected (FC) layer, following the design of [11].

## 5. Experiments

**Implementation Details** For our encoder, we utilize PVTv2-b3 variant that was pre-trained on ImageNet-1K [7] dataset. The network was trained for 50 epochs, with 1e-



(a) G2H+SF (ours) (b) G2H+CamH (c) CamParams [11]

Figure 7. Overall pipelines for our three variants of single view scene scale estimation network.

Table 1. **Quantitative evaluation on samples from public panorama datasets.** P: Panorama dataset (Stanford2D3D, Matterport3D, custom panorama). W: Web image dataset.

| Model | Train Set | Stanford2D3D | | | Matterport3D | | | Web Image | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Angle ↑ | Scale (e-4) ↓ | Height ↓ | Angle ↑ | Scale (e-4) ↓ | Height ↓ | Angle ↑ | Scale (e-2) ↓ | Height ↓ |
| CamParams | P | 0.999 | 1.932 | **0.039** | 0.999 | 4.154 | 0.082 | 0.914 | 21.510 | **0.555** |
| G2H+CamH | P | 0.999 | 1.612 | 0.040 | 0.999 | 3.558 | **0.080** | 0.936 | 21.738 | 0.606 |
| G2H+SF | P | 0.999 | **1.502** | 0.043 | 0.999 | **3.522** | 0.086 | 0.937 | **21.263** | 0.563 |
| G2H+CamH | P + W | 0.999 | 1.924 | **0.036** | 0.999 | 3.702 | **0.081** | 0.938 | 13.785 | 0.331 |
| G2H+SF | P + W | 0.999 | **1.858** | 0.049 | 0.999 | **3.601** | 0.085 | 0.938 | **3.076** | **0.180** |

4 initial learning rate and learning rate decaying to 10% at 15000, 30000, and 60000 iterations. The training took $\sim$12 hours using 8 NVIDIA A100 GPUs, where for each GPU, a batch of 48 images of 256×256 resolution was loaded per iteration. The dataloader is designed to assign equally distributed samples from each dataset in a batch. We directly supervise all the outputs using mean squared error (MSE) loss with equal loss weights. Further implementation details are described in Appendix. Unless specified otherwise, we use the results of our G2H+SF variant for analysis and visualizations.

## 5.1. Model Analysis

For quantitative comparison, we evaluate three metrics, cosine similarity for ground2horizon estimation, RMSE error for scale field prediction, and REL error for camera height. Since G2H+SF model does not estimate camera height directly, averaged value of $SF/\|\text{ground2horizon}\|$ was used for evaluation. Note that camera height estimation performance is reported for completeness, but it may not directly reflect the scene scale estimation. We tested our three models on Stanford2D3D, Matterport3D and web image test sets. Overall results are shown in Tab. 1.

**Direct Parameter Estimation *vs* Scale Field Estimation.** First, we compare our three models, G2H+SF, G2H+CamH, and CamParams, to see the difference between predicting local fields and global parameters. Row 1, 2 and 3 in Tab. 1 show the test results of the three models trained in same training configurations.

In our Stanford2D3D and Matterport3D test set, all three models show nearly perfect results on gravity direction estimation. However, scale metrics show that the performance improves when replacing the global parameter prediction with the equivalent dense field formulation. For example, G2H+CamH is better than CamParams by using the Ground2Horizon field, and G2H+SF is better than G2H+CamH by predicting the scale field. Nonetheless, all three models perform way worse on web image test set, since the set contains out-of-distribution samples regarding camera height.

On the other hand, comparing row 4 and 5, while both G2H+SF and G2H+CamH were trained using the same datasets including our web image dataset, the scale estimation result on web image test set using the G2H+SF model excels the result of G2H+CamH. Considering that the scale field generation using outputs of G2H+CamH exactly follows the scale field formulation we described in Eq. (3), this result indicates that predicting local fields rather than global parameters boosts robustness in scale estimation.

Above observations validate the effectiveness of our proposed approach of representing scene scale as local and dense scale field. Qualitative results using three models are also shown in Fig. 8. G2H+SF shows much reasonable scale estimations results compared to CamParams and G2H+CamH models. Additionally, it is shown that our model implicitly holds the information of the horizon line.

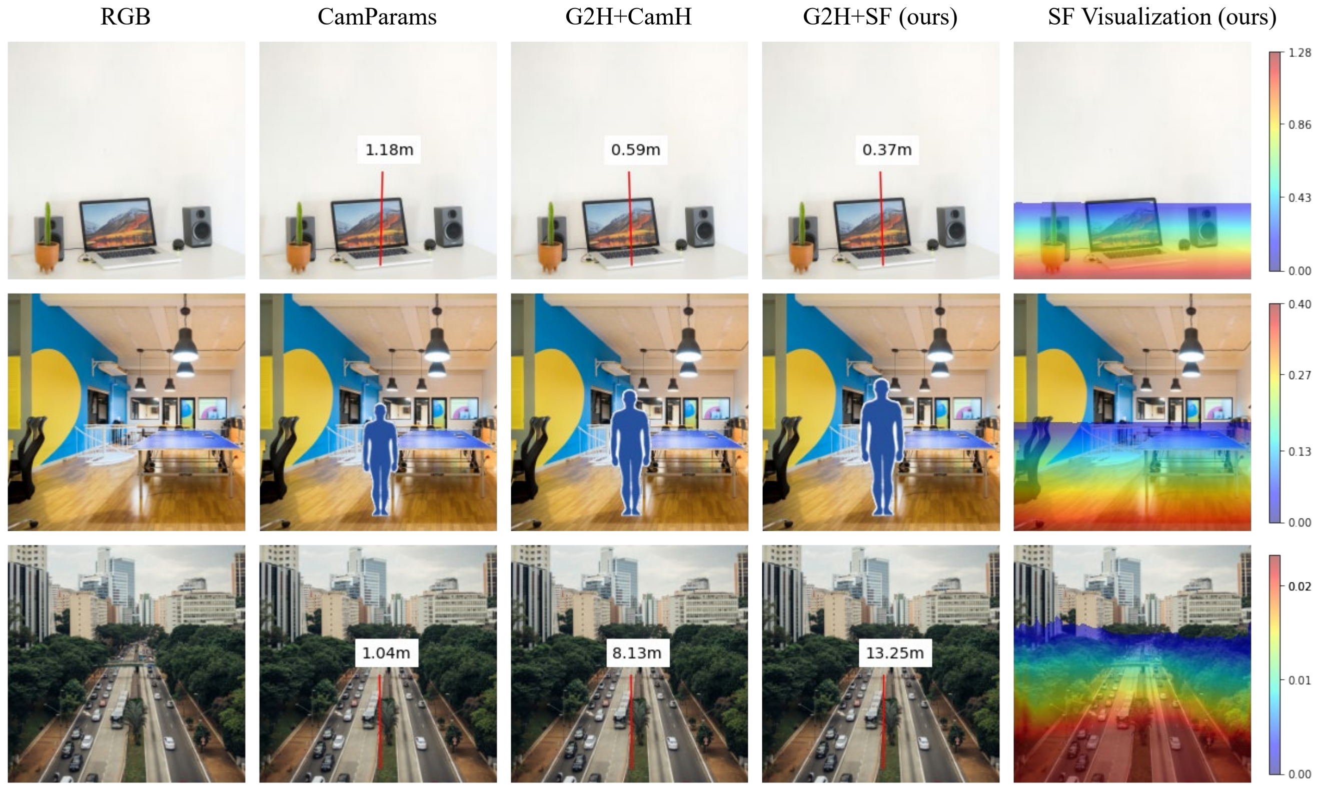


Figure 8. **Qualitative comparison of our models.** Inserted human silhouette set to have height of 1.7m. Metric height measured at pixel coordinate (130, 240) for all examples.

**Effect of Web Image Dataset.** We also ablate the effect of utilizing our web image dataset that covers a wide range of camera heights. In the third and the fifth row in Tab. 1, the same G2H+SF model was used while each was trained on a different set of datasets. In Stanford2D3D and Matterport3D, the network that is only trained on panorama datasets shows better performance, since it was fitted to a certain range of camera heights. However, the result of it on our web image dataset is far behind. We stress the importance of training the network with a huge variety of scenes, thus the importance of our provided image annotation pipeline.

**Generalization From Scene to Object.** We further analyze the robustness of our SF model by processing a sequence of gradually cropped images that show from the overall scene to a certain object. The results are shown in Fig. 9. Both rows show reliable scale estimation results on all of the crops. This indicates that our network estimating scale field works well on both scene-scale and object-centric images.

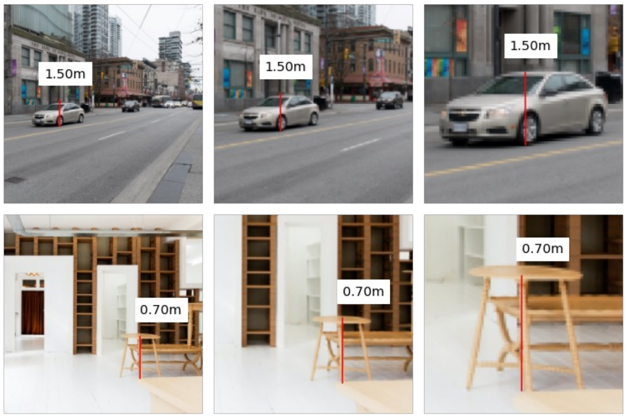
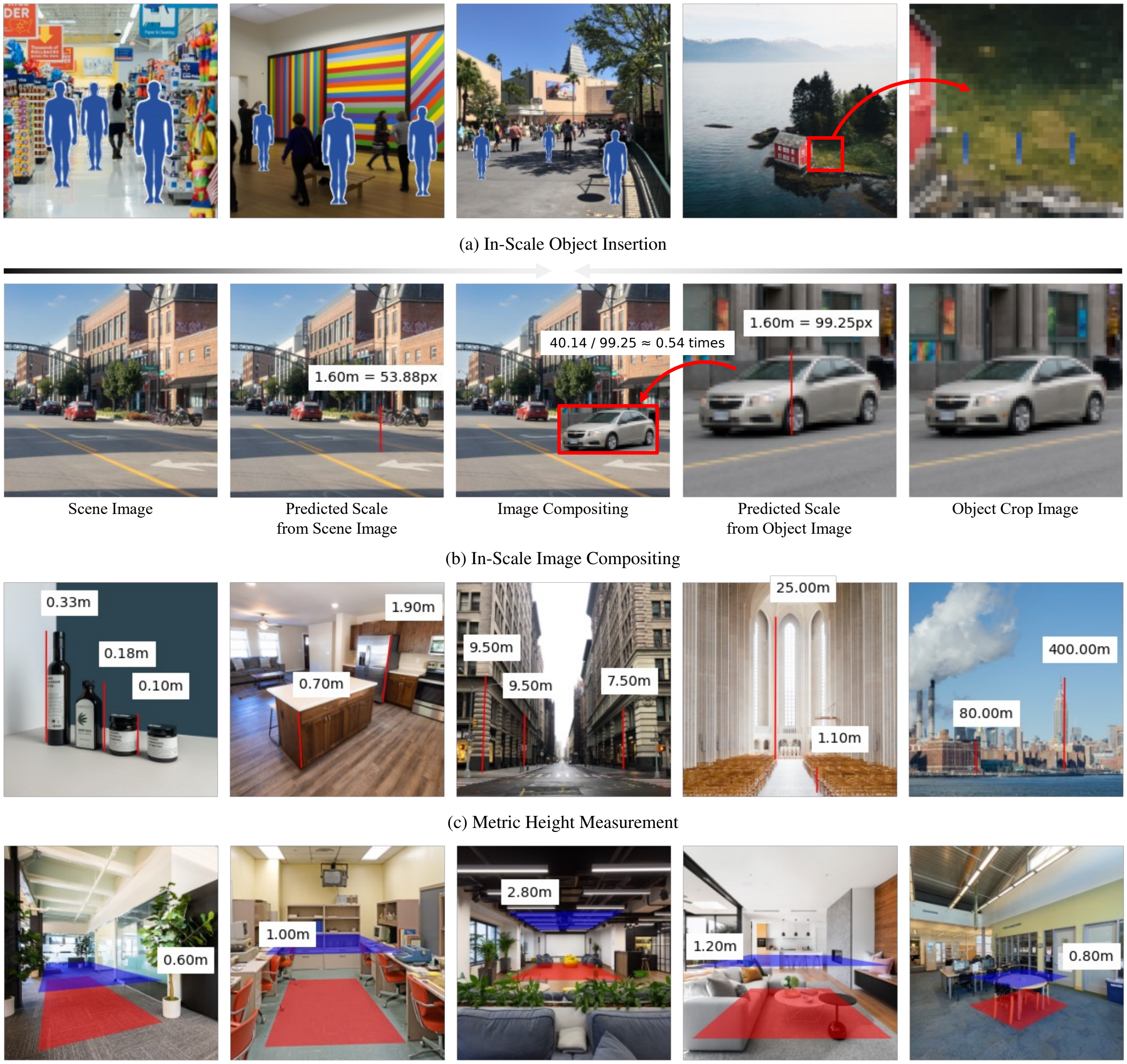


Figure 9. Robustness check by gradually cropping scene image to object-centric image.

## 5.2. Applications

Using our scale field estimation results, we perform scale-aware image editing tasks like object insertion (Fig. 10a) and in-scale image compositing (Fig. 10b), and 3D scene understanding tasks like metric height measurement (Fig. 10c) and ground plane elevation (Fig. 10d). The last example can be utilized for architectural usage or furniture arrangements. The various use cases of the tasks using our scale field predictions are visualized in Fig. 10. Our network can handle very extreme examples with small camera height like the left-most image in Fig. 10c, or with large camera height like the right-most images in Fig. 10a and Fig. 10c.


(a) In-Scale Object Insertion

Scene Image

Predicted Scale from Scene Image

Image Compositing

Predicted Scale from Object Image

Object Crop Image

(b) In-Scale Image Compositing

(c) Metric Height Measurement

(d) Ground Plane Elevation

Figure 10. **Examples of various use cases of our scale field estimations results.** Inserted human silhouette set to have height of 1.7m.

## 6. Conclusion

In this paper, we introduce an effective way to estimate scale information of a scene from a single image. Our scale field, a local, dense and generalizable representation of scale, helps a deep learning network to learn robust scene scale in various range of images, from close-up tabletop images to cityscape images. Also, it enables easy annotation on unlimited images using single-view metrology-based geometry. We believe that our scale field representation and single image scene scale estimation can leverage ambiguous 3D understanding tasks from a single view.

**Limitations.** Our scale field formulation assumes a flat ground plane and is only defined under the horizon line. It loses accuracy and consistency when the image is mainly composed of curved surface, and cannot cover cases like photos that are pointing up so that the horizon line lies under the image boundary. It will be an interesting topic to further upgrade our scale field formulation on even more variety of images, which we aim to focus for our future direction.

## References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 4

[2] Olga Barinova, Victor Lempitsky, Elena Tretiak, and Pushmeet Kohli. Geometric image parsing in man-made environments. In *European conference on computer vision*, pages 57–70. Springer, 2010. 2

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 2

[4] 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. 4

[5] 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. 2

[6] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 1, 2

[7] 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. 5

[8] 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. 2

[9] Jonathan Deutscher, Michael Isard, and John MacCormick. Automatic camera calibration from a single manhattan image. In *European Conference on Computer Vision*, pages 175–188. Springer, 2002. 1, 2, 5

[10] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008. 1, 2, 3

[11] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018. 1, 2, 5, 6

[12] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Amodal completion and size constancy in natural scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 127–135, 2015. 2

[13] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 2

[14] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13916–13925, 2021. 2

[15] Hyunjoon Lee, Eli Shechtman, Jue Wang, and Seungyong Lee. Automatic upright adjustment of photographs with robust camera calibration. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):833–844, 2013. 2

[16] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2

[17] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4521–4530, 2019. 1, 2

[18] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1, 2

[19] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 2

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 1

[21] 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. 2

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 1

[24] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[25] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[26] 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*, pages 234–241. Springer, 2015. 5

[27] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas

Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2

[28] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 5

[29] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. Deepfocal: A method for direct focal length estimation. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1369–1373. IEEE, 2015. 1, 2, 5

[30] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. *arXiv preprint arXiv:1604.02129*, 2016. 1, 2

[31] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[32] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 204–213, 2021. 1

[33] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5657–5665, 2016. 2

[34] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2

[35] Rui Zhu, Xingyi Yang, Yannick Hold-Geoffroy, Federico Perazzi, Jonathan Eisenmann, Kalyan Sunkavalli, and Manmohan Chandraker. Single view metrology in the wild. In *European Conference on Computer Vision*, pages 316–333. Springer, 2020. 1, 2