Single View Metrology in the Wild

Rui Zhu[1 [0000−0002−3266−2514]], Xingyi Yang[1 [0000−0002−1603−9829]], Yannick Hold-Geoffroy[2 [0000−0002−1060−6941]], Federico Perazzi[2 [0000−0002−3636−8267]], Jonathan Eisenmann[2 [0000−0003−2018−0793]], Kalyan Sunkavalli[2 [0000−0002−6030−2348]], and Manmohan Chandraker[1 [0000−0003−4683−2454]]

[1] University of California San Diego, La Jolla CA 92093, USA
{rzhu,x3yang,mkchandraker}@eng.ucsd.edu
[2] Adobe Research, San Jose CA 95110, USA
{holdgeof,perazzi,eisenman,sunkaval}@adobe.com

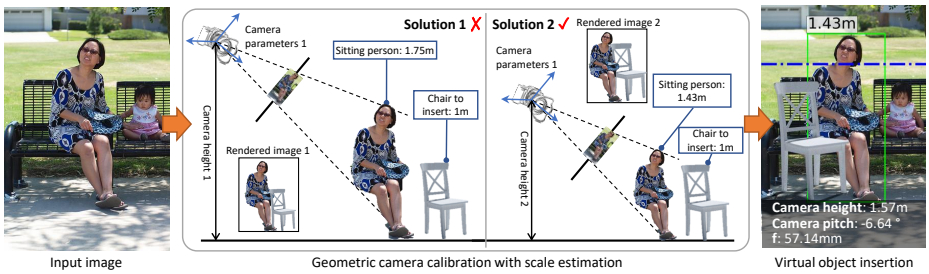

Fig. 1: Given the image on the left, single view metrology can recover the scene and the camera parameters in 3D only up to a global scale factor (for example, the two solutions in the middle). Our method accurately estimates absolute 3D camera parameters and object heights (middle, left) to produce realistic object insertion results (right).

Abstract. Most 3D reconstruction methods may only recover scene properties up to a global scale ambiguity. We present a novel approach to single view metrology that can recover the *absolute* scale of a scene represented by 3D heights of objects or camera height above the ground as well as camera parameters of orientation and field of view, using just a monocular image acquired in unconstrained condition. Our method relies on data-driven priors learned by a deep network specifically designed to imbibe weakly supervised constraints from the interplay of the unknown camera with 3D entities such as object heights, through estimation of bounding box projections. We leverage categorical priors for objects such as humans or cars that commonly occur in natural images, as references for scale estimation. We demonstrate state-of-the-art qualitative and quantitative results on several datasets as well as applications including virtual object insertion. Furthermore, the perceptual quality of our outputs is validated by a user study.

Keywords: Single view metrology, absolute scale estimation, camera calibration, virtual object insertion

1 Introduction

Reconstructing a 3D scene from images is a fundamental problem in computer vision. Despite many successes on this task, most previous works only reconstruct scenes up to an unknown scale. This is true for many problems including structure-from-motion (SfM) from uncalibrated cameras [14], monocular camera calibration in the wild [34,17,35] and single image depth estimation [9,24]. This ambiguity is inherent to the projective nature of image formation and resolving it requires additional information. For example, the seminal work "Single View Metrology" of Criminisi et al. [6] relies on the size of reference objects in the scene.

In this work, we consider the problem of single view metrology "in the wild", where only a single image is available for an unconstrained scene composed of objects with unknown sizes. In particular, we plan to achieve this via geometric camera calibration with absolute scale estimation, *i.e.* recovering camera orientation (alternatively, the horizon in the image), field-of-view, and the *absolute* 3D height of the camera from the ground. Given these parameters, it is possible to convert any 2D measurement in image space to 3D measurements.

Our goal is to leverage modern deep networks to build a robust, automatic single view metrology method that is applicable to a broad variety of images. One approach to this problem could be to train a deep neural network to predict the scale of a scene using a database of images with known absolute 3D camera parameters. Unfortunately, no such large-scale dataset currently exists. Instead, our insight is to leverage large-scale datasets with 2D object annotations [25,36,11,9]. In particular, we make the observation that objects of certain categories such as humans and cars are ubiquitous in images in the wild [25,36] and would make good "reference objects" to infer the 3D scale.

While the idea of using objects of known classes as references to reconstruct camera and scene 3D properties has been used in previous work [16,18], we significantly extend this work by making fewer approximations in our image formation model (*e.g.* full perspective camera vs. zero camera pitch angle, infinite focal length in [16]), leading to better modeling of images in the wild. Moreover, our method learns to predict all camera and scene properties (object and camera height estimation, camera calibration) in an end-to-end fashion; in contrast, previous work relies on individual components that address each sub-task. We demonstrate that this holistic approach leads to state-of-the-art results across all these tasks on a variety of datasets (SUN360, KITTI, IMDB-23K). We also demonstrate the use of our method for applications such as virtual object insertion, where we may automatically create semantically meaningful renderings of a 3D object with known dimensions (see Fig. 1).

In summary, we propose the following contributions:

- A state-of-the-art Single View Metrology method for images in the wild that performs geometric camera calibration with absolute scale—horizon, field-of-view, and 3D camera height—from a monocular image.
- A weakly supervised approach to train the above method with only 2D bounding box annotations by using an in-network image formation model.
- Application to scale-consistent object insertion in unconstrained images.

2 Related Work

Camera calibration To estimate the camera parameters, numerous efforts have been made for estimating camera intrinsics [34,4,8,17] by explicit reasoning or learning in a data-driven fashion. In addition, to estimate camera extrinsics, *e.g.* camera rotation angles or in the form of horizon estimation, classical methods [38,3,7,22] look for low-level features such as line segments. More recently, methods are proposed to directly regress the horizon from the input image [26,35,20] by learning from large-scale datasets annotated with ground truth horizons. The human sensitivity to calibration errors is studied in [17].

Depth prediction in the wild As we discussed in Section 1, the problem of scene scale estimation will be solved if we are able to predict pixel-wise depth for the scene. There has been a line of work in this topic. For domains which we can acquire the ground truth absolute depth with depth sensors we may learn to predict depth in a supervised fashion [9,32,19]. However given the limitation of the range or mobility of the sensors, these datasets are more or less limited to specific scenes. In other cases, people are able to acquire ground truth from stereo matching [10,12,33] but a large-scale stereo depth dataset for images in the wild is still absent. Other people have turn to proxy methods for collecting depth via structure-from-motion (SfM) [24,23], or in the form of relative depth [5], or from synthetic images [27,2]. However, these methods either produce depth without absolute scale, or pose a domain gap to natural images.

Single view metrology Another line of work that seeks to estimate 3D scene parameters from images is Single View Metrology [6], which recovers scene structure in 3D from purely 2D measurements. These methods look for 2D properties such as vanishing lines and vanishing points as well as object locations, to establish relations among 3D sizes of objects in the image based on 2D measurements. Some works have been done to embed Single View Metrology in a framework to estimate the size of an unknown object in the scene or the camera height itself [1,18,16,21], given at least one reference object with known size.

3 Method

3.1 Recovering 3D Parameters from 2D Annotations

We start by describing the image formation model that allows us to associate 3D camera parameters, 3D object sizes (*i.e.* heights) and 2D bounding boxes. This is also illustrated in Fig. 2.

We assume the world is composed of a dominant ground plane on which all objects are situated, and a camera that observes the scene. We adopt a perspective camera model similar to [16,17], which is parameterized by camera angles (yaw φ, pitch θ and roll ψ), focal length f and camera height h_{cam} to the ground (see Fig. 2). For the measurements in the vertical axis of image frame,

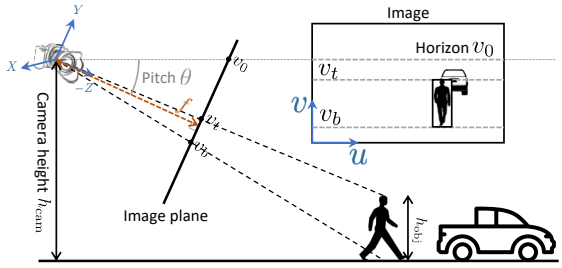

Fig. 2: Camera model of the scene (bottom) and measurements in image space (top).

the location of the horizon is v_0, while the vertical image center is at v_c. Each object bounding box have a top v_t and bottom v_b location in the image. We assume all images were taken with zero roll, or were rectified beforehand [22]. We further assume, without loss of generality, a null yaw and zero distortion from rectification. Camera pitch θ can be computed from v_c, v_0 and f using

$$\theta = \arctan \frac{v_c - v_0}{f}. \tag{1}$$

Consider a thin object of height h_{obj} with its bottom located at $[x, 0, z]^T$ in 3D and its top at $[x, h_{\text{obj}}, z]^T$. These points project to $[u, v_b]^T$ and $[u, v_t]^T$ respectively in the camera coordinates. Based on the perspective camera model (see Eqn. 3 of the supplementary material), we have

$$v_t = \frac{(f\cos\theta + v_c \sin\theta)h_{\text{obj}} \left\|\cos\theta\right\| + (-f\sin\theta + v_c\cos\theta)z - f h_{\text{cam}}}{h_{\text{cam}} \left\|\cos\theta\right\| \tan\theta + z\cos\theta}, \tag{2}$$

$$v_b = \frac{-f\sin\theta z + v_c\cos\theta z + f^2 h_{\text{cam}}}{z\cos\theta}, \tag{3}$$

where $[u_c, v_c] \in \mathbb{R}^2$ is the camera optical center which we assume is known. Substituting z from Eqn. 3 into Eqn. 2, we may derive v_t from camera focal length f, pitch θ, camera height h_{cam} and object height h_{obj}. Hoeim et al. [16] make a number of approximations including $\cos\theta \approx 1$, $\sin\theta \approx \theta$ and $(v_c - v_0) \times (v_c - v_0)/f^2 \approx 0$, to linearly solve for the object height:

$$h_{\text{obj}} = h_{\text{cam}} \frac{v_t - v_b}{v_0 - v_b}. \tag{4}$$

In contrast, we model the full expression accounting for all the camera parameters.

Eqn. 2 and Eqn. 3 establish a relationship between the camera parameters, including its 3D height, the 3D heights of objects in the scene, and the 2D projections of these objects into the image. Moreover, note that once we can estimate these parameters, we can directly infer the 3D size of *any* object from its 2D bounding box (Eqn. 4), thus resolving the scale ambiguity in monocular reconstruction. In the following section, we introduce ScaleNet, a deep network that leverages this constraint to create weak supervision to learn to predict 3D camera height.

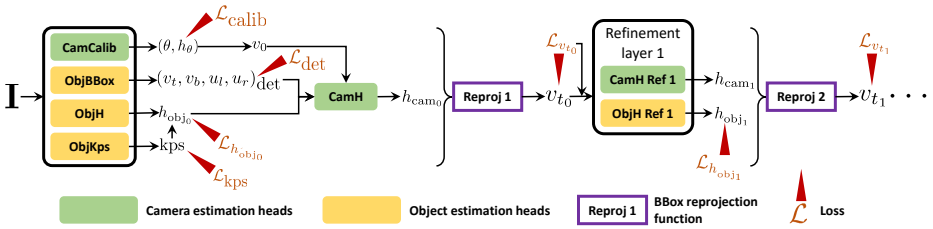

Fig. 3: Overview of our method. From the input image **I**, the camera calibration module estimates pitch θ and field of view h_θ, and the object estimation modules estimate keypoints for person, object heights h_{obj} and bounding boxes. The estimated horizon v_0, bounding boxes and object heights are fed into the camera height estimation module to give an initial estimation h_{cam}. The bounding box reprojection errors $\mathcal{L}_{v_t}$ are then computed from the reprojection module (see Eqn. 2 and Eqn. 5), and together with other variables are fed to the refinement network to estimate updates on the camera height and object heights. Several layers of refinement are made to produce the final estimation.

3.2 ScaleNet: Single View Metrology Network with Absolute Scale Estimation

Previous work [16,18] has shown that when scene parameters (*e.g.* camera parameters, object sizes) are reasonable, reprojected 2D bounding boxes should ideally fit the detected ones in the image frame. We follow a similar path in our weakly-supervised learning framework and specifically focus on humans and cars, given that they are the most commonly occurring object categories in datasets of images in the wild (*e.g.* COCO dataset [25]).

Our end-to-end method, referred to as **ScaleNet (SN)**, is split into two parts, which we describe in Fig. 3. First, all the object bounding boxes and camera parameters except camera height are jointly estimated by a geometric camera calibration network. These parameters are directly supervised during training. Second, a cascade of PointNet-like networks [31] estimates and refines the camera height (scene scale) based on the previous outputs. This second part is weakly supervised at each stage using a bounding box reprojection loss.

Camera calibration module and object heads The camera calibration module is inspired by [17], where we replace their backbone with Mask R-CNN [15,28], to which we add heads to estimate the camera parameters. To train the camera calibration module, we follow the representation of [17], including bins and training loss. However, instead of predicting the focal length f (in pixels), we find it easier to predict the vertical field of view h_θ, which can be converted to the focal length and the horizon midpoint using:

$$f = \frac{\frac{1}{2}h_{\text{im}}}{\tan\left(\frac{1}{2}h_\theta\right)}, \quad v_0 = \frac{\frac{1}{2}h_{\text{im}}\tan\theta}{\tan\left(\frac{1}{2}h_\theta\right)} + \frac{h_{\text{im}}}{2},$$

where h_{im} is the height of the image in pixels. We also use additional heads to estimate the object bounding box, height and person keypoints (since we find that a person's 3D height in an image is closely related to their pose) from ROI features which share the same backbone as the camera estimation module. Please refer to the supplementary material for the full architecture of this network. In total, we enforce three losses on this part of the model, *i.e.* the camera calibration loss $\mathcal{L}_{\text{calib}}\big(\theta, h_\theta\big)$ and the detection losses $\mathcal{L}_{\text{det}}$ and $\mathcal{L}_{\text{kps}}$.

The i_{th} object with detected 2D top position $v^i_{t_{\text{det}}}$ is reprojected to the image by Eqn. 2 to v^i_t with estimated object height, camera height and camera parameters, and we define the bounding reprojection error as

$$\mathcal{L}_{v_t}\left(\{v^i_t\}^N_{i=1}\right) = \frac{1}{N}\sum_{i=1}^{N}\left\|v^i_{t_{\text{det}}} - v^i_t\right\|. \tag{5}$$

Object height prior The above bounding box supervision has the same scale ambiguity as previous work. However, explicitly modeling 3D object heights allows us to use a prior on size to regularize the network to produce a meaningful object height estimation. We follow [16] and use a Gaussian prior fit from statistics (*i.e.* for 1.70±0.09m for people and 1.59±0.21m for cars). For an object i of height h^i_{obj} and prior Gaussian distribution $\mathcal{P}(x;\mu,\sigma)$, we define the height prior loss as

$$\mathcal{L}_{h_{\text{obj}}}\left(\{h^i_{\text{obj}}\}^N_{i=1}\right) = -\frac{1}{N}\sum_{i=1}^{N}\mathcal{P}(h^i_{\text{obj}};\mu,\sigma). \tag{6}$$

Camera height estimation module Directly predicting camera height from images would require the network to learn to be robust to a wide variety of appearance properties (object, layout, illumination, *etc.*). Instead, we design a camera height estimation module that leverages the strong *geometric* relationship between camera height, 2D bounding boxes and the horizon. As exemplified in Fig. 4, both images are composed of a group of standing people while the horizons are not fully visible in the back. At first glance, both images seem to have the same camera orientation, since the people take roughly the same space in the image. However the camera height is quite different between both images. Based on this observation, instead of estimating camera height from image appearance, we take advantage of middle level representations of the scene (*e.g.* object bounding boxes and estimated horizon line) and feed those to the camera height estimation module which is derived from PointNet [30]. Its input is the concatenation of all object bounding box coordinates and the offset between the bounding box and horizon, *i.e.* $\gamma_0 = [v_0, u_{l_{\text{det}}}, u_{r_{\text{det}}}, v_{t_{\text{det}}}, v_{b_{\text{det}}}, v_{t_{\text{det}}} - v_0, v_{b_{\text{det}}} - v_0, h_{\text{obj}}]^T \in \mathbb{R}^8$ where $u_{l_{\text{det}}}$ and $u_{r_{\text{det}}}$ are the left and right coordinates of the detected bounding box. The network outputs the camera height as a discrete probability distribution. Finally, a weighted sum after a `softmax` is applied to obtain the camera height estimation.

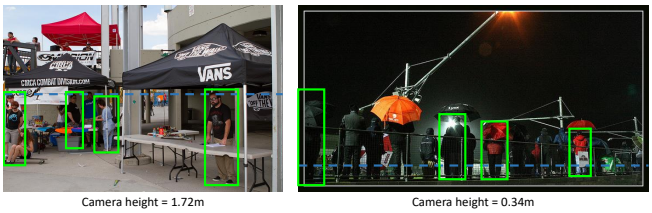

Fig. 4: Example of images with different camera heights exhibiting similar bounding boxes.

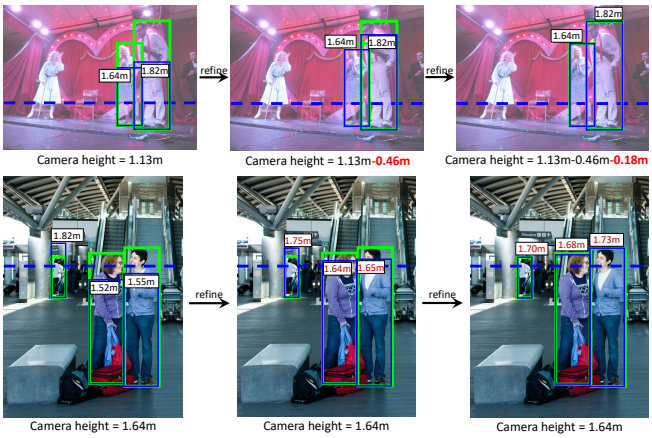

Fig. 5: Example of cascade refinements of camera height (top) and person heights (bottom). The refined parameters are labelled in red.

Cascade refinement layers We observe that we can iteratively refine the camera height by considering all scene parameters jointly. Inspired by the cascade refinement scheme from [31], we propose to look at the error residual—in our case the object bounding box reprojection error—and predict a *difference* to the estimated parameters. The whole process is highlighted in Fig. 5, where the reprojected object bounding boxes are shorter than the detected ones at first. After a first step of refinement, the network reduces the camera height to reduce the object bounding boxes error, and so on. To this end, we design layers of refinement, where in layer $j \in \{1, 2, \dots, M\}$ a camera height and object height refinement module takes as input the object bounding box reprojection residuals and the other camera parameters, formally as $\gamma_j = [v_0, (u_l^i, u_r^i, v_{t_{j-1}}^i, v_b^i)_{\text{det}}, v_{t_{j-1}}^i - v_{t_{\text{det}}}^i, h_{\text{obj}_{j-1}}^i, h_{\text{cam}_{j-1}}]^T \in \mathbb{R}^8$ where $i \in \{1, 2, \dots N\}$ is the object index. Each refinement layer j predicts updates Δh_{cam_j} and $\Delta h_{\text{obj}_j}^i$ so that $h_{\text{obj}_j}^i = \Delta h_{\text{obj}_j}^i + h_{\text{obj}_{j-1}}^i$ and $h_{\text{cam}_j} = \Delta h_{\text{cam}_j} + h_{\text{cam}_{j-1}}$. An object bounding box reprojection loss $\mathcal{L}_{v_{t_j}}$ and object height prior $\mathcal{L}_{h_{\text{obj}_j}}$ are enforced for each layer.

The final training loss is a weighted combination of the losses, written as

$$\mathcal{L}\left(\{\{v_{l_j}^i\}_{i=1}^N, \{h_{\text{obj}_j}{}^i\}_{i=1}^N\}_{j=0}^M, \theta, f, h_\theta, (...)_{\text{det}}, \text{kps}\right) = \alpha_1 \sum_{j=1}^M \mathcal{L}_{v_{l_j}} + \alpha_2 \sum_{j=0}^M \mathcal{L}_{h_{\text{obj}_j}} + \alpha_3 \mathcal{L}_{\text{calib}} + \alpha_4 \mathcal{L}_{\text{det}} + \alpha_5 \mathcal{L}_{\text{kps}}, \quad (7)$$

where $\alpha_{1..5}$ are weighting constants to balance the losses during training.

3.3 Datasets

In the following, we describe the datasets and their preprocessing used for training and evaluation. Data generation details, statistics, sampled visualization of all datasets can be found in the supplemental material.

Calib: Camera calibration dataset To train the camera calibration module to estimate camera pitch and field of view from a single image, we follow the data generation pipeline from [17], where data are cropped from the SUN360 database [37] of 360° panoramas with sampled camera parameters. We split the resulting camera calibration dataset into 397,987 images for training and 2,000 images for validation. For simplicity, we refer to this dataset as the `Calib` dataset.

COCO-Scale: Scale estimation dataset from COCO While the `Calib` dataset provides a large and diversified dataset of images and many ground truth camera pitch and field of view parameters, it does not provide camera height. To complement this dataset, we use the COCO dataset [25], which allows us to train our method in a weakly-supervised way. This dataset features 2D annotations of object bounding boxes, keypoints of person, and stuff annotation. We further extend these annotations by using Mask R-CNN [15] to infer objects of certain categories, *e.g.* person and car. These additional annotations complement the ones provided in the dataset, which together form our candidate object set. We refer to this dataset as the `COCO-Scale` dataset.

We filter out invalid objects which do not satisfy our scene model, using the stuff annotation (*e.g.* ground, grass, water) to infer the support relationship of an object with its surrounding pixels; we only keep objects that are most likely situated on a plane (*e.g.* people standing on grassland, cars on a street). For the *person* category, we use the detected keypoints from Mask R-CNN [28] to detect people with both head and ankles visible to ensure the obtained bounding box is amodal as in [18]. We further filter the images based on aspect ratio, object size and number of objects to keep bounding boxes of certain shape.

This pruning step yields 10,547 training images, 2,648 validation images, taken from COCO's `train2017` and `val2017` splits respectively. We further obtain test test images from `val2017` and ensure no overlap exists between the splits. We call this person-only subset `COCO-Scale-person`. For the multi-category setting (`COCO-Scale-mulCat`), we look for images including both cars and people, which provides us with 12,794 images for training, 3,189 for validation, and 584 for testing.

Fig. 6: (Left) Annotated person bounding box (red) with ground truth height and detected person (green) with keypoints (colored). (Right) Calculation of upright ratio.

KITTI We use the KITTI [11] dataset to evaluate our camera and object height estimations. We apply the same filtering rules as used on COCO, yielding 298 images for person-only setting (`KITTI-person`), and 234 images for the multi-category setting (`KITTI-mulCat`).

IMDB-23K celebrity dataset for person height evaluation IMDB-23K [13] is a collection of online images of celebrities, with annotations of body height from the IMDB website. We use this dataset to evaluate our object height prediction. However, these height annotations may not be exact and we treat them as *pseudo* ground truth to draw comparisons. We apply the same filtering rules as on COCO, and the filtered dataset consists of 2,550 test images with one celebrity labelled with height in each image. An image from this dataset is shown in Fig. 6.

4 Experiments

4.1 Baseline methods

We use Hoiem *et al.* [16] as the baseline method. For fair comparison, we employ our object proposals or top predictions from Mask R-CNN [28] as input to this method to replace the original detector [29], which enhances the original method. We set up 2 baseline models: (1) *PGM*: the original model based on a Probabilistic Graphical Model, which takes in object proposals and surface geometry, and predicts camera height and horizon. Object heights can be computed from Eqn. 4 by directly minimizing the reprojection error; (2) *PGM-fixedH*: same as PGM but assumes all objects have canonical height. For people, we use 1.7m which is the mean of the person height prior used in [16]). For cars we use 1.59m.

4.2 Training

We train our model in two stages. Firstly, we train the camera calibration network with full supervision from the Calib dataset using camera calibration losses and the detection & keypoint estimation heads with full supervision from COCO ground truth with losses following [28]. The backbone, camera calibration head,

Table 1: $\mathcal{L}_{v_t}$ on COCO-Scale-person

		PGM-fixedH	SN-L1	SN-L3	SN-L3-mulCat	SN-L3-kps-can
$\mathcal{L}_{v_t}$	mean	0.1727	0.1502	0.0717	0.0613	**0.0540**
	std.	0.3598	0.2579	0.1875	0.1874	0.1677
	med.	0.0793	0.0693	0.0116	**0.0074**	0.0094

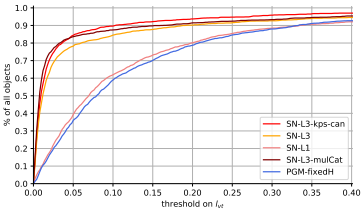

Fig. 7: $\mathcal{L}_{v_t}$ of all objects on COCO-Scale-person under varying thresholds.

and detection & keypoint heads are initialized with a pre-trained Mask R-CNN model [28].

For the second stage, the object height estimation module is trained together while other modules are finetuned, in a weakly-supervised fashion, with the full loss in Eqn. 7. Training details can be found in the supplemental material.

Variants of ScaleNet (SN) We evaluate several variations of ScaleNet (*SN*): (1) *SN-L1*: one layer architecture with direct prediction of object height and camera height without refinements. (2) *SN-L3*: one layer for initial prediction with 2 additional refinement layers. (3) *SN-L3-mulCat*: same as SN-L3 but with objects of multiple categories as input and trained on COCO-Scale-mulCat. (4) *SN-L3-kps-can*: same as SN-L3 but training keypoints prediction and predicting each person in upright height instead of actual height. An upright ratio computed as $l_{\text{actual}}/l_{\text{upright}}$ in Fig. 6 is an approximation of the actual ration in 3D that takes into account the person's pose. It is multiplied to the predicted upright height to obtain actual height, and the height prior is applied to the predicted upright height.

Training results on COCO-Scale We calculate the bounding reprojection error from Eqn. 2 on the test split which is an indication of 2D bounding box fits. The results are shown in Table 1 and Fig. 7.

4.3 Evaluation on COCO-Scale-person

Since we do not have ground truth 3D annotations on the `COCO-Scale-person` dataset, we evaluate performance using a user study on virtual object insertion. We evaluate the plausibility of our estimates on the resulting scale and perspective effects of an inserted object via an A/B test on `COCO-Scale-person`. In our evaluation, we render a 1m height chair alongside each object. For each of the 4 pairs of models, we insert the chair in 50 random test images. 10 users for each pair of models were asked to choose which image of the pair is more realistic w.r.t. the scales of all chairs.

Fig. 8: Scene parameters estimation and virtual object insertion results on COCO-Scale. The detected boxes are shown in green and reprojected ones in blue. The horizon is shown as dashed blue line. Camera parameters are overlaid on the top (camera height as y_c, focal length in millimeters as f_{mm} assuming 35mm full-frame sensor, and pitch as θ). A chair of 1m tall is inserted alongside each person with the estimated parameters.

As can be seen in Table 2, our results improve when adding multiple categories (*SN-L3-mulCat*) or regress keypoint and account for the pose while computing person height (*SN-L3-kps-can*). The best variant *SN-L3-kps-can* outperforms other methods significantly. This is consistent with the bounding box reprojection error in Table 1 where *SN-L3-kps-can* is the best performing method.

Fig. 8 shows a qualitative evaluation of our method. The first row shows the benefit of using the upright ratio, leading to better estimations in *SN-L3-kps-can*. The second row shows our method displays better behavior in cases of multiple

Table 2: A/B test results for scale on COCO-Scale-person

	SN-L3	SN-L1	SN-L3	SN-L3-kps-can	SN-L3	SN-L3-mulCat	SN-L3-kps-can	PGM
preference	**54.6%**	43.8%	42.8%	**57.2%**	**50.7%**	49.3%	**59.5%**	40.5%

Table 3: Evaluation errors on all *pedestrian* objects of KITTI-person. PGM-fixedH assumes a canonical object height, while PGM explicitly solves for the object height that minimizes bounding box reprojection error $\mathcal{L}_{v_t}$ (ideally to zero), as a result of which the $\mathcal{L}_{v_t}$ errors of PGM are grayed out as they are not directly comparable to others. Best results in bold (lower is better).

		PGM-fixedH	PGM	SN-L1	SN-L3	SN-L3-kps-can	SN-L3-mulCat
input	car person	✓	✓	✓	✓	✓	✓
$\mathcal{E}_{h_{\text{obj}}}$	mean	0.0863	0.1358	**0.0837**	0.0956	0.1014	0.0849
	std.	0.0570	0.1406	0.0610	0.0751	0.0864	0.0714
	med.	0.0800	0.0916	0.0727	0.0770	0.0864	**0.0685**
$\mathcal{L}_{v_t}$	mean	0.0767	0.0016	0.0980	0.0331	0.0585	**0.0283**
	std.	0.0638	0.0009	0.1000	0.0644	0.0815	0.0415
	med.	0.0618	0.0003	0.0724	0.0128	0.0815	**0.0127**
$\mathcal{E}_{h_{\text{cam}}}$	mean	0.1408		0.2356	0.1988	0.2649	**0.1264**
	std.	0.1585		0.2860	0.3162	0.3207	0.1147
	med.	0.1096		0.1821	0.1160	0.1666	**0.0878**

objects with diverse heights. In the third row, we demonstrate robustness to outliers (the person on the bus).

4.4 Quantitative evaluation on KITTI

Since KITTI provides ground truth for all of the parameters our method estimate, we can directly evaluate the errors in bounding box reprojection $\mathcal{L}_{v_t}$, camera height estimation $\mathcal{E}_{h_{\text{cam}}}$ and object height estimation $\mathcal{E}_{h_{\text{obj}}}$ as shown in Table 3 and Table 4 on KITTI-person and KITTI-mulCat respectively, where

$$\mathcal{E}_{h_{\text{cam}}} = \left\| h_{\text{cam}} - h_{\text{cam}_{\text{gt}}} \right\|, \mathcal{E}_{h_{\text{obj}}} = \frac{1}{N} \sum_{i=1}^{N} \left\| h_{\text{obj}} - h_{\text{obj}_{\text{gt}}} \right\| \tag{8}$$

h_{cam} and h^i_{obj} are the final estimated camera height and height of object i, and $h_{\text{cam}_{\text{gt}}}$ and $h^i_{\text{obj}_{\text{gt}}}$ are their ground truth values respectively.

Our method SN-L3-mulCat outperforms previous work on both KITTI-person and KITTI-mulCat. This method takes into account the cues from multiple categories to perform inference, giving it an advantage on scenes with high-diversity content (see Table 4). Qualitative results are shown in Fig. 9. Please refer to the supplementary material for more results.

Fig. 9: Scene parameters estimation results with SN-L3-mulCat on KITTI-mulCat. Reprojected pedestrians are in blue, while cars are in magenta.

Table 4: Evaluation errors on all *pedestrian* objects of KITTI-mulCat following specifications of Table 3

		PGM	PGM	SN-L3	SN-L3-mulCat	SN-L3-mulCat
input	car		✓			✓
	person	✓	✓	✓	✓	✓
$\mathcal{E}_{h_{\text{obj}}}$	mean	0.1198	0.1177	0.1266	0.1092	**0.0956**
	std.	0.1007	0.0932	0.0994	0.0883	0.0811
	med.	0.0896	0.0939	0.0968	0.0876	**0.0780**
$\mathcal{L}_{v_t}$	mean	0.0008	0.0008	0.0647	**0.0303**	0.0712
	std.	0.0013	0.0011	0.1124	0.0465	0.1153
	med.	0.0003	0.0004	0.0166	**0.0123**	0.0297
$\mathcal{E}_{h_{\text{cam}}}$	mean	0.1379	0.1519	0.3464	0.1547	**0.1222**
	std.	0.1735	0.1676	0.3693	0.1687	0.1235
	med.	**0.0703**	0.1096	0.2278	0.0991	0.0904

4.5 Quantitative evaluation on IMDB-23K

IMDB-23K provides annotation for the registered height of a person, which we assume is the height of the person standing straight (the upright height). Since all of our models except SN-L3-kps-can predict the actual person height (influenced by the specific pose, viewpoint, bounding box drawing, etc.), we use the upright ratio (see Fig. 6) computed from detected keypoints to convert the actual height back to upright height. This upright ratio allows us to compute upright height from all methods,and compare against the pseudo ground truth, as included in Table 5 and Fig. 10. Since the ground truth annotations are only valid for standing people, we further get a subset of the test set where the estimated upright ratio from keypoint prediction is less than 0.90, which typically denotes a non-standing person. Table 6 evaluates the methods on this subset and shows that SN-L3-kps-can, which directly accounts for the upright ratio in training and

Table 5: $\mathcal{E}_{h_{\text{obj}}}$ and $\mathcal{L}_{v_t}$ on IMDB-23K following specifications of Table 3

		PGM-fixedH	PGM	SN-L1	SN-L3	SN-L3-kps-can	SN-L3-mulCat
$\mathcal{E}_{h_{\text{obj}}}$	mean	0.0843	0.2234	**0.0832**	0.0891	0.1003	0.0990
	std.	0.0638	0.2246	0.0688	0.0818	0.0920	0.0915
	med.	0.0700	0.1644	0.0706	**0.0695**	0.0920	0.0777
$\mathcal{L}_{v_t}$	mean	0.0983	0.0157	0.1011	0.0431	0.0920	**0.0416**
	std.	0.0546	0.0185	0.1706	0.1324	0.1257	0.1537
	med.	0.0689	0.0105	0.0441	0.0071	0.1257	**0.0056**

Table 6: $\mathcal{E}_{h_{\text{obj}}}$ on IMDB-23K (non-standing person).Best results in bold (lower is better)

		PGM	SN-L3	SN-L3-kps-can
$\mathcal{E}_{h_{\text{obj}}}$	mean	0.1552	0.1591	**0.1212**
	std.	0.1379	0.1788	0.1072
	med.	0.1177	0.1013	**0.0909**

Fig. 10: Scene parameters estimation results with SN-L3-kps-can on IMDB-23K.

inference, performs better in getting upright heights compared to other models; visual comparisons are shown in Fig. 10 and the supplementary material.

5 Conclusion and Future Work

We present a learning-based method that performs geometric camera calibration with absolute scale from images in the wild. We demonstrate that our method provides state-of-the-art results on multiple datasets. Despite this advance, our method is hindered by some limitations.

Our single dominant ground plane assumption does not always hold in the wild. Urban scene may provide multiple supporting surfaces at different heights (tables, balconies), so objects may not be laying on the assumed ground plane. Also, the ground may be non-flat in nature environments.

Our method is highly biased on *appearance*. Adding amodal reasoning, as proposed in [18], would be an interesting way forward to perform holistic scene reasoning. We would like to tackle these limitations as future work.

References

1. Andaló, F.A., Taubin, G., Goldenstein, S.: Efficient height measurements in single images based on the detection of vanishing points. Computer Vision and Image Understanding **138**, 51–60 (2015)
2. Atapour-Abarghouei, A., Breckon, T.P.: Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2800–2810 (2018)
3. Barinova, O., Lempitsky, V., Tretiak, E., Kohli, P.: Geometric image parsing in man-made environments. In: European conference on computer vision. pp. 57–70. Springer (2010)
4. Chen, Q., Wu, H., Wada, T.: Camera calibration with two arbitrary coplanar circles. In: European Conference on Computer Vision. pp. 521–532. Springer (2004)
5. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in neural information processing systems. pp. 730–738 (2016)
6. Criminisi, A., Reid, I., Zisserman, A.: Single view metrology. International Journal of Computer Vision **40**(2), 123–148 (2000)
7. Denis, P., Elder, J.H., Estrada, F.J.: Efficient edge-based methods for estimating manhattan frames in urban imagery. In: European conference on computer vision. pp. 197–210. Springer (2008)
8. Deutscher, J., Isard, M., MacCormick, J.: Automatic camera calibration from a single manhattan image. In: European Conference on Computer Vision. pp. 175–188. Springer (2002)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. p. 2366–2374 (2014)
10. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7628–7637 (2019)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
13. Gunel, S., Rhodin, H., Fua, P.: What face and body shapes can tell us about height. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
14. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (2003)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
16. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision **80**(1), 3–15 (2008)
17. Hold-Geoffroy, Y., Sunkavalli, K., Eisenmann, J., Fisher, M., Gambaretto, E., Hadap, S., Lalonde, J.F.: A perceptual measure for deep single image camera calibration. In: CVPR. pp. 2354–2363 (2018)
18. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Amodal completion and size constancy in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 127–135 (2015)
19. Kim, W., Ramanagopal, M.S., Barto, C., Yu, M.Y., Rosaen, K., Goumas, N., Vasudevan, R., Johnson-Roberson, M.: Pedx: Benchmark dataset for metric 3-d

pose estimation of pedestrians in complex urban intersections. IEEE Robotics and Automation Letters **4**(2), 1940–1947 (2019)
20. Kluger, F., Ackermann, H., Yang, M.Y., Rosenhahn, B.: Temporally consistent horizon lines. In: 2020 International Conference on Robotics and Automation (ICRA) (2020)
21. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. ACM transactions on graphics (TOG) **26**(3), 3 (2007)
22. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs with robust camera calibration. IEEE transactions on pattern analysis and machine intelligence **36**(5), 833–844 (2013)
23. Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., Freeman, W.T.: Learning the depths of moving people by watching frozen people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4521–4530 (2019)
24. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
26. Man, Y., Weng, X., Li, X., Kitani, K.: Groundnet: Monocular ground plane estimation with geometric consistency. In: Computer Vision and Pattern Recognition (CVPR) (2018)
27. Martinez II, M.A.: Beyond Grand Theft Auto V for Training, Testing and Enhancing Deep Learning in Self Driving Cars. Ph.D. thesis, Princeton University (2018)
28. Massa, F., Girshick, R.: maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark` (2018), accessed: [Oct 16, 2019]
29. Murphy, K.P., Torralba, A., Freeman, W.T.: Graphical model for recognizing scenes and objects. In: NIPS. pp. 1499–1506 (2003)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
31. Ranftl, R., Koltun, V.: Deep fundamental matrix estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 284–299 (2018)
32. Wang, C., Miguel Buenaposada, J., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2022–2030 (2018)
33. Wang, L., Shen, X., Zhang, J., Wang, O., Lin, Z., Hsieh, C.Y., Kong, S., Lu, H.: Deeplens: shallow depth of field from a single image. arXiv preprint arXiv:1810.08100 (2018)
34. Workman, S., Greenwell, C., Zhai, M., Baltenberger, R., Jacobs, N.: Deepfocal: A method for direct focal length estimation. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 1369–1373. IEEE (2015)
35. Workman, S., Zhai, M., Jacobs, N.: Horizon lines in the wild. arXiv preprint arXiv:1604.02129 (2016)
36. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE winter conference on applications of computer vision. pp. 75–82. IEEE (2014)

37. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2695–2702. IEEE (2012)
38. Zhai, M., Workman, S., Jacobs, N.: Detecting vanishing points using global image context in a non-manhattan world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5657–5665 (2016)