2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019
[Award version is not citable, please cite the interactive session version]

Geo-Supervised Visual Depth Prediction

Xiaohan Fei, Alex Wong, and Stefano Soatto

***Abstract*— We propose using global orientation from inertial measurements, and the bias it induces on the shape of objects populating the scene, to inform visual 3D reconstruction. We test the effect of using the resulting prior in depth prediction from a single image, where the normal vectors to surfaces of objects of certain classes tend to align with gravity or be orthogonal to it. Adding such a prior to baseline methods for monocular depth prediction yields improvements beyond the state-of-the-art and illustrates the power of gravity as a supervisory signal.**

I. Introduction

The visual world is heavily affected by gravity, including the shape of many artifacts such as buildings and roads, and even natural objects such as trees. Gravity provides a globally consistent orientation reference that can be reliably measured with low-cost inertial sensors present in mobile devices from phones to cars. We call a machine learning system able to exploit global orientation, *geo-supervised*. Gravity can be easily inferred from inertial sensors without the need for dead-reckoning, and the effect of biases is negligible in the context of our application.

To measure the influence of gravity as a supervisory signal, we choose the extreme example of predicting depth from a single image. This is, literally, an impossible task in the sense that there are infinitely many three-dimensional (3D) scenes that can generate the same image. So, any process that yields a point estimate has to rely heavily on priors. We call the resulting point estimate a *hypothesis*, or *prediction*, and use public benchmark datasets to quantitatively evaluate the improvement brought about by exploiting gravity. Of course, only certain objects have a shape that is influenced by gravity. Therefore, our prior has to be applied *selectively*, in a manner that is informed by the semantics of the scene.

Our approach to geo-supervised Visual Depth Prediction is based on training a system end-to-end to produce a map from a single image and an estimate of the orientation of gravity in the (calibrated) camera frame to an inverse depth (disparity) map. In one mode of operation, the training set uses calibrated and rectified stereo pairs, together with a semantic segmentation module, to evaluate a loss function differentially on the images where geo-referenced objects are present. In a second mode, we use monocular videos instead and minimize the reprojection (prediction) error. Optionally, we can leverage modern visual-inertial odometry (VIO) and mapping systems that are becoming ubiquitous from hand-held devices to cars.

The key to our approach is a prior, or regularizer, that selectively biases certain regions of the image that correspond to geo-referenced classes such as roads, buildings, vehicles, and trees. Specifically, points in space that lie on the surface of such objects should have normals that either align with, or are orthogonal to, gravity. This is in addition to standard regularizers used for depth prediction, such as left-right consistency and piecewise smoothness.

While at training time a semantic segmentation map is needed to apply our prior selectively, it is never passed as input to the network. Therefore, at test time it is not needed, and an image is simply mapped to the disparity.

The ultimate test for a prior is whether it helps improve end-performance. To test our prior, we first incorporated it into two top-performing methods, one binocular (Sect. V-B) and one monocular (Sect. V-C), in the KITTI benchmark [6], and showed consistent performance improvement in all metrics. To further challenge our prior, we took two other baselines which were not the top performers. We then added our prior and tested the results against the top performers in the latest benchmark. We also performed generalizability tests (Sect. V-E), ablation studies (Sect. V-D) and demonstrated our approach with VIO on hand-held devices (Sect. V-F).

II. Related work

Early learning-based depth prediction approaches [11], [13], [25], [26] predict depth using local image patches and then refine it using Markov random fields (MRFs). Recent works [3], [16] leverage deep networks to directly learn a representation for depth prediction where the networks are typically based on the multi-scale fully convolutional encoder-decoder structure. These methods are fully supervised and do not generalize well outside the datasets on which they are trained. Latest self-supervised methods [5], [7], [37] have shown better performance on benchmarks with better generalization.

There is a large body of work [20], [31], [34], [35] on self-supervised monocular depth prediction following Godard *et al*. [7] and Zhou *et al*. [37], which simply use the reprojection error as a learning criterion, as has been customary in 3D reconstruction for decades. Generic priors such as piecewise smoothness and left-right consistency are also encoded into the network as additional loss terms. Our work is in-line with these self-supervised approaches, but we also exploit class-specific regularizers beyond the generic ones.

This work was supported by ONR N00014-17-1-2072 and ARO W911NF-17-1-0304.

The authors are with the Computer Science Department, University of California, Los Angeles, USA. Email: {feixh, alexw, soatto}@cs.ucla.edu

In terms of exploiting the relation of different geometric quantities in an end-to-end learning framework, closely related works include [17], [23], [32], where surface normals are explicitly computed by using either a network [32] or some heuristics [23]. While the former is computation intensive, the latter relies on heuristics and thus is sub-optimal. In contrast, by using losses proposed in this paper, we directly regularize depth via the depth-gravity relation without a separate surface normal predictor. Besides, both [32] and [17] are supervised, while ours is self-supervised with the photometric loss and guided by global orientation and the semantics of the scene.

Earlier work on semantic segmentation [27] relied on local features, and have been improved by incorporating global context using various structured prediction techniques [14], [24]. Starting from the work of Long *et al*. [19], fully convolutional encoder-decoder networks have been a staple in semantic segmentation. Although we do not address semantic segmentation, we leverage per-pixel semantic labeling enabled by existing systems to aid depth prediction in the form of providing class-specific priors and an attention mechanism to selectively apply such priors, which is different from joint segmentation and depth prediction approaches [10].

The idea of using class-specific priors to facilitate reconstruction is not new [8], [15]. In [8], class-specific shape priors in the form of spatially varying anisotropic smoothness terms are used in an energy minimization framework to reconstruct small objects. Though promising, this system does not scale well. An efficient inference framework [14] has been used with a CRF model over a voxel-grid to achieve real-time performance by [15]. While all these methods explore class-specific priors in various ways, none has used them in an end-to-end learning framework. Also, all the methods above take range images as inputs, which are then fused with semantics during optimization, while ours exploits semantics at an earlier stage – when generating such range images which themselves can serve as priors for dense reconstruction and other inference tasks.

III. Methodology

In this section, we introduce our loss functions as regularizers added to existing models at training time, in addition to data terms (photometric loss) and generic regularizers (smoothness loss). We dub our loss semantically informed geometric loss (SIGL) because geometric constraints are selectively applied to certain image regions, where a semantic segmentation module informs the selection. Fig. 1 illustrates part of our training diagram. In Sect. III-C, we review baseline models used in our experiments and show that the application of our losses on top of them improves performance (Sect. V).

A. Semantically informed geometric loss

During training, we assume to be given a partition of the image plane into semantic classes $c \in C$ that have a consistent geometric correlate. For instance, a pixel with image coordinates $(x,y) \in \mathbb{R}^2$ and class $c(x,y) =$ "road" is often associated to a normal plane oriented along the vertical direction (direction of gravity), whereas $c =$"building" has a normal vector orthogonal to it. We also assume we are given the calibration matrix K of the camera capturing the images, so the pixel coordinates (x,y) on the image plane back-project to points in space via

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} Z(x,y) \tag{1}$$

where $Z(x,y)$ is the depth Z of the point along the projection ray determined by (x,y).

Any subset $\Omega \subset \mathbb{R}^2$ of the image plane that is the image of a spatial plane with normal vector $N \in \mathbb{R}^3$, at distance $\|N\|$ from the center of projection, satisfies a constraint of the form $\mathbf{X}_i^T N = 1$ for all i, assuming the plane does not go through the optical center. Stacking all the points into matrix $\bar{\mathbf{X}} \doteq [\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_M]^\top$, we have $\bar{\mathbf{X}} N = \mathbf{1}$, where $\mathbf{1}$ is a vector of M ones, and $M = |\Omega|$ is the cardinality of the set Ω. If the direction, but not the norm, of the vector N is known, a scale-invariant constraint can be easily obtained by removing the mean of the points, so that (details in Sect. III-B)

$$(\mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}} N = 0. \tag{2}$$

The scale-invariant constraint above can be used to define a loss to penalize deviation from planarity:

$$L_{HP}(\Omega_{HP}) = \frac{1}{|\Omega_{HP}|} \|(\mathbf{I} - \frac{1}{|\Omega_{HP}|}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}}\gamma\|_2^2 \tag{3}$$

where N in Eq. (2) is replaced by normalized gravity γ due to the homogeneity of Eq. (2), and the squared norm is taken assuming the network predicts per-pixel depth $Z(x,y)$ up to additive zero-mean Gaussian noise. $\Omega_{HP} \subset \mathbb{R}^2$ is a subset of the image plane whose associated semantic classes have horizontal surfaces, such as "road", "sidewalk", "parking lot", etc. We call this loss "horizontal plane" loss, where the direction of gravity γ can be reliably and globally estimated.

Similarly, a "vertical plane" loss can be constructed to penalize deviation from a vertical plane whose normal N has *both unknown direction and norm* but lives in the null space of γ, *i.e.,* $N \in \mathcal{N}(\gamma)$. Thus, the vertical plane loss reads

$$L_{VP}(\Omega_{VP}) = \min_{\substack{N \in \mathcal{N}(\gamma) \\ \|N\|=1}} \frac{1}{|\Omega_{VP}|} \|(\mathbf{I} - \frac{1}{|\Omega_{VP}|}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}} N)\|_2^2 \tag{4}$$

where the constraint $\|N\| = 1$ avoids trivial solutions $N = 0$ again due to the homogeneity of the objective; Ω_{VP} is a subset of the image plane whose associated semantic classes have vertical surfaces, such as "building", "fence", "billboard", etc. The constrained minimization problem in the vertical plane loss L_{VP} is due to the unknown direction of the surface normals and introduces some difficulties in training. We discuss approximations in Sect. III-B.

B. Explanation of the objectives

Our idea is essentially to use priors about surface normals to regularize depth prediction. An intuitive way to achieve

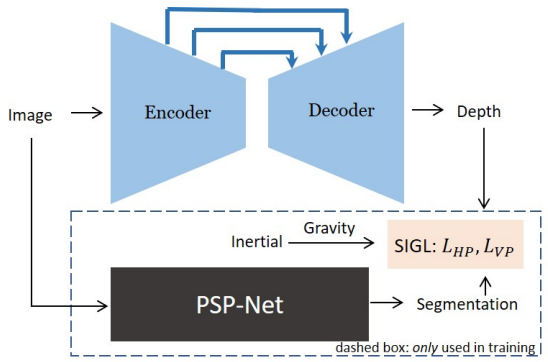

Fig. 1: *Illustration of geo-supervised visual depth prediction.* Our visual depth predictor is an encoder-decoder convolutional neural network with skip connections. At inference time, the network takes an RGB image as the only input and outputs an inverse depth map. At training time, gravity extracted from inertial measurements biases the depth prediction *selectively*, which is informed by semantic segmentation produced by PSPNet. The other identical stream of the network and the photometric losses used for training are omitted in this figure.

this is to compute the surface normals from the depth values first and then impose regularity, which will eventually bias the depth predictor via backpropagation. However, such a method involves normal estimation from depth, which can be problematic, especially with a simplistic but noisy normal estimator [23].[1] On the other hand, one could train a deep network to compute surface normals [32], which is costly. Therefore, *we do not compute surface normals but directly regularize the depth values* via the scale-invariant constraint Eq. (2) which is a function of depth and the direction of gravity.

In what follows, we give an explanation of L_{HP} Eq. (3) from a statistical perspective. Let $M = |\Omega_{HP}|$ to avoid notation clutter and expand Eq. (3):

$$(\mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}}\gamma \tag{5}$$

$$= \begin{bmatrix} 1-\frac{1}{M} & \cdots & -\frac{1}{M} \\ \vdots & \ddots & \vdots \\ -\frac{1}{M} & \cdots & 1-\frac{1}{M} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1^\top\gamma \\ \mathbf{X}_2^\top\gamma \\ \cdots \\ \mathbf{X}_M^\top\gamma \end{bmatrix} = \begin{bmatrix} \vdots \\ \left(\mathbf{X}_i - \frac{1}{M}\sum_{j=1}^M \mathbf{X}_j\right)^\top\gamma \\ \vdots \end{bmatrix} \tag{6}$$

Let $\mu = \frac{1}{M}\sum_{j=1}^M \mathbf{X}_j$ be the sample mean of the 3D coordinates and the horizontal plane loss L_{HP} reads

$$L_{HP}(\Omega_{HP}) = \frac{1}{M}\sum_{i=1}^{M}\left((\mathbf{X}_i - \mu)^\top\gamma\right)^2 \tag{7}$$

which is the sample variance of the 3D coordinates projected to the direction of gravity γ (coinciding with the surface normal for horizontal planes). To minimize L_{HP} is to minimize the variance of the 3D coordinates along the surface normal.

[1]For instance, one can compute the point-wise surface normal as the cross product of two vectors tangent to the surface, where the tangent vectors are approximated by connecting the underlying point to its nearest neighbors on the surface.

Similarly, to minimize L_{VP} Eq. (4) is to minimize the variance of the 3D coordinates along some direction perpendicular to gravity. However, if the direction is unknown, one needs to jointly solve the direction while minimizing L_{VP}, which explains the constrained quadratic problem in L_{VP}. Though this can be solved via eigendecomposition, the gradients of the solver – needed in backpropagation – are non-trivial to compute. In fact, representing an optimization procedure as a layer of a neural network is an open research problem [1]. To alleviate both numerical and implementation difficulties, we uniformly sample unit vectors from the null space of gravity and compute the minimum of the objective over the samples as an approximation to the loss. Empirically, we found using eight directions sampled every 45 degrees from 0 to 360 generally performs well.

C. View synthesis as supervision and baselines

To showcase the ability to improve upon existing self-supervised monocular depth prediction networks, we add our losses to two publicly available models – Godard [7] (`LR-Consistency`) and Yin [34] (`GeoNet`) – as baselines and perform both quantitative and qualitative comparisons. We additionally apply our losses to Zhan [35] (`Stereo-Temporal`) and Wang [31] (`DDVO`), the state-of-the-art methods in their respective training setting, stereo pairs/videos, and monocular videos. `LR-Consistency` is trained with rectified stereo image pairs, `GeoNet` and `DDVO` use monocular videos while `Stereo-Temporal` uses stereo videos. At test time, all training settings result in a system that takes a single image as input and predicts an inverse depth map as output. We show that by applying our losses to the baselines `LR-Consistency` and `GeoNet`, we achieve better performance than the state-of-the-art methods `Stereo-Temporal` and `DDVO`. Furthermore, we produce new state-of-the-art results by applying our losses to `Stereo-Temporal` and `DDVO`.

1) Training with stereo pairs: At training time, our first baseline model (`LR-Consistency`) takes a single left image as its input and predicts two disparity maps $D^L, D^R : \mathbb{R}^2 \supset \Omega \to \mathbb{R}_+$ for both left and right cameras. The network follows the fully convolutional encoder-decoder structure with skip connections. The total loss consists of three terms: Appearance loss, smoothness of disparity and left-right consistency, each of which is evaluated on both the left and the right streams across multiple scale levels. Here we address the view synthesis loss, which serves as the data term and is part of the appearance loss:

$$L_{\text{vs}}^L = \frac{1}{|\Omega|}\sum_{(x,y)\in\Omega} \|I^L(x,y) - I^R(x + D^L(x,y), y)\|_1. \tag{8}$$

The view synthesis loss is essentially the photometric difference of the left image $I^L(x,y)$ and the right image warped to the left view $I^R(x + D^L(x,y), y)$ according to the left disparity prediction $D^L(x,y)$. The right view synthesis loss is constructed in the same way. Though only one disparity map is needed at inference time, it has been shown that predicting

both left and right disparity maps and including the left-right consistency loss Eq. (9) are in general beneficial [7].

$$L_{\text{lr}}^{L} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \|D^{L}(x,y) - D^{R}(x + D^{L}(x,y), y)\|_1 \quad (9)$$

2) Training with stereo videos: In our second baseline `Stereo-Temporal`, stereo videos are used to train a monocular depth predictor, where two frames of a stereo pair and another frame one time step ahead are involved in constructing a stereo-temporal version of the photometric loss: For the stereo pair, Eq. (8) is applied while for the temporal pair, Eq. (10) (detailed below) is applied.

3) Training with monocular videos: To train our third and fourth baseline models (`GeoNet` and `DDVO`), a single reference frame I_t is fed into the depth network and frames $I_{t'}, t' \in W_t$ in a temporal window centered at t are used to construct the view synthesis loss, also known as reprojection error:

$$L_{\text{vs}} = \frac{1}{|W_t||\Omega|} \sum_{t'\in W_t} \sum_{(x,y)\in\Omega} \|I_t(x,y) - I_{t'}\big(\pi(\hat{g}_{t't}\mathbf{X})\big)\|_1 \quad (10)$$

which is the difference between the reference frame I_t and neighboring frames $I_{t'}$ warped to it. $\mathbf{X}$ is the back-projected point defined in Eq. (1), π is a central (perspective) projection, and $\hat{g}_{t't}$ is the relative camera pose up to an unknown scale predicted by an auxiliary pose network which takes both I_t and $I_{t'}$ as its input. Note that the pose and depth networks are coupled via the view synthesis loss at training time; at test time, the depth network alone is needed to perform depth prediction with a single image as its input. Interestingly, in Sect. V-F we found that replacing the pose network with pose estimation from VIO produces better results compared to the multi-task learning diagram where pose and depth networks are trained simultaneously, which sheds light on the use of classic SLAM/Odometry systems in developing better learning algorithms.

A detailed discussion about other losses serving as regularization terms is beyond the scope of this paper and can be found in [7], [31], [34], [37].

IV. Implementation Details

A. Semantic segmentation

At training time, we use PSPNet [36] pre-trained on the CityScapes dataset [2] provided by the authors to obtain per-pixel labeling. For every pixel $(x,y) \in \mathbb{R}^2$, a probability distribution over 19 classes is predicted by PSPNet, of which the most likely class $c(x,y) \in C$ determines the orientation of the surface where the back-projected point $\mathbf{X}$ sits. We group the 19 classes into 7 categories[2] according to the CityScapes benchmark and test our losses on all of them. Empirically, we found that it is most beneficial to apply our losses to the "flat", "vehicle" and "construction" categories and therefore all the comparisons on KITTI against baseline methods are made with these categories regularized. The influence of other categories is studied in Sect. V-D.

[2]"flat": road, sidewalk; "human": rider, person; "vehicle": car, truck, bus, train, motorcycle, bicycle; "construction": building, wall, fences; "object": pole, traffic light, traffic sign; "nature": vegetation, terrain; "sky": sky.

B. Gravity

For imagery captured by a static platform equipped with an inertial measurement unit (IMU), one can use the gravity $\gamma_b \in \mathbb{R}^3$ measured in the body frame (coinciding with the IMU frame) and simply apply the body-to-camera rotation $R_{cb} \in \mathrm{SO}(3)$ to obtain the gravity in the camera frame $\gamma = R_{cb}\gamma_b$ which is then used in Eq. (3) and (4). For moving platforms, one resorts to robust VIO, which is well studied [22], [30]. In Sect. V-F, we demonstrated our approach on a visual-inertial odometry dataset, where both camera pose and gravity are estimated online by VIO.

For our experiments on the KITTI dataset, thanks to the GPS/IMU sensor package which provides linear acceleration of the sensor platform measured both in the body frame ($\alpha_b \in \mathbb{R}^3$) and the spatial frame ($\alpha_s \in \mathbb{R}^3$), we are able to compute the spatial-to-body rotation $R_{bs} \in \mathrm{SO}(3)$ and then bring the gravity $\gamma_s = [0,0,9.8]^\top$ from the spatial frame to the camera frame $\gamma = R_{cb}R_{bs}\gamma_s$. In all settings, R_{cb} (rotational part of the body-to-camera transformation) is obtained via offline calibration procedures.

C. Training details

A GTX 1080 Ti GPU and Adam [12] optimizer are used in our experiments. Depending on different model variants and input image sizes, training time varies from 8 hours to 16 hours. For `LR-Consistency` and `GeoNet` which were initially implemented in TensorFlow, we implemented our losses also in TensorFlow and applied them to the existing code bases. Code of `Stereo-Temporal` is available online, but in Caffe, thus we migrated their model to TensorFlow and applied our losses. We also implemented our losses in PyTorch, which were then applied to `DDVO` of which the PyTorch version was made available by the author. Our code is available at `https://github.com/feixh/GeoSup`.

V. Experiments

To enable quantitative evaluation, we exploit the KITTI benchmark, and test our approach against the state-of-the-art as described in detail below (Sect. V-B&V-C). We also carried out ablation studies (Sect. V-D) and tested the generalizability of our approach (Sect. V-E). In addition to KITTI, which features planar motion in driving scenarios, we have conducted experiments on VISMA dataset [4] – an indoor visual-inertial odometry dataset captured under non-trivial ego-motion (Sect. V-F).

A. KITTI Eigen split

We compare our approach with recent state-of-the-art methods on the monocular depth prediction task using the KITTI Eigen split [3] in two training domains: stereo pairs/videos and monocular videos (Sect. III-C). The Eigen split test set contains 697 test images selected from 29 of 61 scenes provided by the raw KITTI dataset. Of the remaining 32 scenes containing 23,488 stereo pairs, 22,600 pairs are

TABLE I: Error and Accuracy Metrics

Metric	Definition
AbsRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert}{Z^{\text{gt}}(x,y)}$
SqRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert^2}{Z^{\text{gt}}(x,y)}$
RMSE	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert^2}$
RMSE log	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log Z(x,y)-\log Z^{\text{gt}}(x,y)\rvert^2}$
$\log_{10}$	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log Z(x,y)-\log Z^{\text{gt}}(x,y)\rvert$
Accuracy	% of $Z(x,y)$ s.t. $\delta \doteq \max\left(\frac{Z(x,y)}{Z^{\text{gt}}(x,y)}, \frac{Z^{\text{gt}}(x,y)}{Z(x,y)}\right) <$ threshold

$Z(x,y)$ is the predicted depth at $(x,y)\in\Omega$ and $Z^{\text{gt}}(z,y)$ is the corresponding ground truth. Three different thresholds ($1.25, 1.25^2$ and 1.25^3) are used in the accuracy metric as a convention in the literature.

used for training, and the rest is used for validation per the training split proposed by [5]. To generate ground truth depth maps for validation and evaluation, we take the Velodyne data points associated with each image and project them from the Velodyne frame to the left RGB camera frame. Each resulting ground truth depth map covers approximately 5% of the corresponding image and may be erroneous. To handle this, first, we use the cropping scheme proposed by [5], which masks out the potentially erroneous extremities from the left, right and top areas of the ground truth depth map. Then we evaluate depth prediction only at pixels where ground truth depth is available. For visualization, we linearly interpolate each sparse depth map to cover the entire image (Fig. 2).

We additionally provide quantitative evaluations of variants of the models pre-trained on CityScapes and fine-tuned on KITTI. CityScapes dataset contains 22,973 training stereo pairs captured in various cities across Germany with a similar modality as KITTI. We cropped each input image to keep only the top 80% of the image, removing the reflective hood.

The error and accuracy metrics, which are initially proposed by [3] and adopted by others, are used (Table I). Also as a convention in the literature, performances evaluated with depth prediction capped at 50 and 80 meters are reported as suggested by [7]. The choice of 80 meters is two-fold: 1) maximum depth present in the KITTI dataset is on the order of 80 meters and 2) non-thresholded measures can be sensitive to the significant errors in depth caused by prediction errors at small disparity values. For the same reason, depth prediction is capped at 70 meters in the Make3D experiment. Prediction capped at 50 meters is also evaluated since depth at closer range is more applicable to real-world scenarios.

B. Training with stereo pairs

The first baseline we adopt is Godard [7] (with `VGG` [29] as feature extractor), to which SIGL is imposed at training time along with the view synthesis loss Eq. (8) and other generic regularizers used in [7]. The model is trained from scratch with stereo pairs following the Eigen split and compared to both supervised [3], [18] and self-supervised methods [7], [35]. In addition, we apply our losses to variants of the baseline (with `ResNet` [9] as feature extractor; w/ & w/o post-processing) and evaluate different training schemes (w/ & w/o pre-training on CityScapes). Quantitative comparisons can be found in Table II, where the results with SIGL added as an additional regularizer follow the results of the baseline models and variants. In the column marked "Data", `K` refers to Eigen split benchmark on the KITTI dataset, and `CS` refers to the CityScapes dataset. Methods marked with `CS+K` are pre-trained on CityScapes and then fine-tuned on KITTI Eigen split. `pp` denotes post-processing. Cap Xm means depth predictions are capped at X meters. Results of Zhan [35] `Stereo-Temporal` are taken from their paper. The rest of the results are taken from [7] unless otherwise stated.

We want to remind the reader that the first baseline model atop which we built ours is Godard [7] `VGG` which initially performed worse than the `Stereo-Temporal` model of Zhan [35] by a large margin, but by applying our losses to the baseline at training time we managed to boost its performance and make it perform even better than the `Stereo-Temporal` model at test time. Note that the `Stereo-Temporal` model also exploits temporal information in addition to stereo pairs for training while our first baseline built atop Godard does not.

As a second baseline, we apply our losses additionally to the `Stereo-Temporal` model of Zhan to further push the state-of-the-art. Table II shows that our losses improve the `Stereo-Temporal` model across all error metrics with the accuracy metrics $\delta < 1.25^2$ and $\delta < 1.25^3$ being comparable. Another variant of Zhan's model pre-trains on NYU-V2 [28] in a fully supervised fashion and is therefore not pertinent to this comparison. Fig. 2 shows a head-to-head qualitative comparison of ours and the baseline models.

TABLE II: Training with stereo pairs on KITTI.

Method	Data	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Depth: cap 80m								
TrainSetMean*	K	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen [3] `Coarse`*	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen [3] `Fine`*	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu [18]*	K	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard [7] `VGG`	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
+SIGL	K	**0.139**	**1.211**	**5.702**	**0.239**	**0.816**	**0.928**	**0.966**
Zhan [35] `Stereo-Temporal`	K	0.144	1.391	5.869	0.241	0.803	0.928	0.969
+SIGL	K	**0.137**	**1.061**	**5.692**	**0.239**	**0.805**	0.928	0.969
Godard [7] `VGG pp`	CS+K	0.124	1.076	5.311	0.219	0.847	0.942	0.973
+SIGL	CS+K	**0.114**	**0.885**	**4.877**	**0.203**	**0.858**	**0.950**	**0.978**
Godard [7] `ResNet pp`	CS+K	0.114	0.898	4.935	0.206	0.861	0.949	0.976
+SIGL	CS+K	**0.112**	**0.836**	**4.892**	**0.204**	**0.862**	**0.950**	**0.977**
Depth: cap 50m								
Garg [5]	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard [7] `VGG`	K	0.140	0.976	4.471	0.232	0.818	0.931	0.969
+SIGL	K	**0.132**	**0.891**	**4.312**	**0.225**	**0.831**	**0.936**	**0.970**
Zhan [35] `Stereo-Temporal`	K	0.135	0.905	4.366	0.225	0.818	0.937	0.973
+SIGL	K	**0.131**	**0.829**	**4.217**	**0.224**	**0.824**	0.937	0.973
Godard [7] `VGG pp`	CS+K	0.112	0.680	3.810	0.198	0.866	0.953	0.979
+SIGL	CS+K	**0.108**	**0.658**	**3.728**	**0.192**	**0.870**	**0.955**	**0.981**
Godard [7] `ResNet pp`	CS+K	0.108	0.657	3.729	0.194	0.873	0.954	0.979
+SIGL	CS+K	**0.106**	**0.615**	**3.697**	**0.192**	**0.874**	**0.956**	**0.980**

* With ground truth depth supervision.
+SIGL: training with SIGL enabled

C. Training with monocular videos

To demonstrate the effectiveness of our loss in the second training setting (monocular videos), we impose SIGL to our

third (Yin [34]) and fourth (Wang [31]) baseline. Using the KITTI Eigen split, we follow the training and validation 3-frame sequence selection proposed by [37] where the first and third frames are treated as the source views and the central (second) frame is treated as the reference as in Eq. (10). Of the 44,540 total sequences, 40,109 are used for training and 4,431 for validation. We evaluate our system on the aforementioned 697 test images [3]. The same training and evaluation scheme are also applied to other top-performing methods [20], [37] in addition to the selected baselines.

Table III shows detailed comparisons against state-of-the-art self-supervised methods trained using monocular video sequences. We compare against best-performing model variants of Wang [31] (`PoseCNN` & `PoseCNN+DDVO`) and Yin [34] (`ResNet`) with and without pre-training on CityScapes. By adding our losses to existing models, we observe systematic performance improvement across all metrics. Though initially performing worse than Wang [31] `PoseCNN+DDVO`, Yin [34] `ResNet` with the proposed losses even outperforms the original `PoseCNN+DDVO`. Moreover, we achieve new state-of-the-art by adding our losses to `PoseCNN+DDVO` trained on both CityScapes and KITTI. Fig. 2 illustrates representative image regions where we do better.

TABLE III: Training with monocular videos on KITTI.

Method	Data	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Depth: cap 80m								
Zhou [37]	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Mahjourian [20]	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Yin [34] `ResNet`	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
+SIGL	K	**0.142**	**1.124**	**5.611**	**0.223**	**0.813**	**0.938**	**0.975**
Wang [31] `PoseCNN`	K	0.155	1.193	**5.613**	0.229	0.797	0.935	0.975
+SIGL	K	**0.147**	**1.076**	5.640	**0.227**	**0.801**	0.935	0.975
Wang [31] `PoseCNN+DDVO`	K	0.151	1.257	5.583	0.228	**0.810**	0.936	0.974
+SIGL	K	**0.146**	**1.068**	**5.538**	**0.224**	0.809	**0.938**	**0.975**
Zhou [37]	CS+K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian [20]	CS+K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yin [34] `ResNet`	CS+K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
+SIGL	CS+K	**0.147**	**1.076**	**5.468**	**0.222**	**0.806**	**0.938**	**0.976**
Wang [31] `PoseCNN+DDVO`	CS+K	0.148	1.187	5.496	0.226	0.812	0.938	0.975
+SIGL	CS+K	**0.142**	**1.094**	**5.409**	**0.219**	**0.821**	**0.941**	**0.976**
Depth: cap 50m								
Zhou [37]	K	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Mahjourian [20]	K	0.155	0.927	4.549	0.231	0.781	0.931	0.975
Yin [34] `ResNet`	K	0.147	0.936	4.348	0.218	0.810	0.941	0.977
+SIGL	K	**0.135**	**0.834**	**4.193**	**0.208**	**0.831**	**0.948**	**0.979**
Wang [31] `PoseCNN`†	K	0.149	0.920	4.303	0.216	0.813	0.943	0.979
+SIGL	K	**0.140**	**0.816**	**4.234**	**0.212**	**0.818**	**0.945**	**0.980**
Wang [31] `PoseCNN+DDVO`†	K	0.144	0.935	4.234	0.214	**0.827**	0.945	0.977
+SIGL	K	**0.139**	**0.808**	**4.180**	**0.209**	0.826	**0.948**	**0.980**
Zhou [37]	CS+K	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Mahjourian [20]	CS+K	0.151	0.949	4.383	0.227	0.802	0.935	0.974
Yin [34] `ResNet`*	CS+K	/	/	/	/	/	/	/
+SIGL	CS+K	**0.141**	**0.837**	**4.160**	**0.209**	**0.823**	**0.947**	**0.980**
Wang [31] `PoseCNN+DDVO`†	CS+K	0.142	0.901	4.202	0.213	0.827	0.946	0.978
+SIGL	CS+K	**0.135**	**0.832**	**4.119**	**0.206**	**0.836**	**0.949**	**0.980**

* Not available. † Evaluated with prediction released by the author.
+SIGL: training with SIGL enabled

D. Ablation study

To study the contribution of each semantic category to the performance improvement, we performed an ablation study: We apply our losses to different semantic categories, one at a time, train the network until convergence, and show how the quality of depth prediction varies (Table IV). In Table IV, Godard *et al*. [7] is the baseline model where only the most generic regularizers, *e.g.,* smoothness and consistency, are used. The second column indicates the semantic category of which the depth prediction is regularized using our losses in addition to the generic regularizers. For the meaning of the semantic categories, see Sect. IV-A.

It turns out that the "flat" category contributes most to the performance gain over the baseline model, which is expected because most of the KITTI images contain a large portion of roads and sidewalks. We also observed that regularization of the "construction" and "vehicle" category provides reasonable improvement while the "nature" category (trees and hedges) helps a little. Applying our priors to the "human", "sky" and "object" categories does not consistently improve over the baseline, for the following reasons: "sky" does not have well-defined surface normals; "human" has deformable surfaces of which normals can point arbitrarily; "object" category consists of thin structures which project to few pixels rendering it hard to apply segmentation and our losses. The best is achieved when we apply our losses to "vehicle", "construction" and "flat" categories, denoted by `V+C+F` in Table IV.

TABLE IV: Ablation study on KITTI.

Method	Category	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Godard [7]	/	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	`Human`	0.152	1.394	5.945	0.251	0.801	0.921	0.963
Ours	`Sky`	0.148	1.368	5.864	0.245	0.807	0.923	0.964
Ours	`Object`	0.146	1.335	5.986	0.249	0.800	0.920	0.963
Ours	`Nature`	0.146	1.292	5.826	0.247	0.804	0.923	0.964
Ours	`Vehicle`	0.143	1.304	5.797	0.241	0.814	0.927	0.966
Ours	`Construction`	0.142	1.252	5.729	0.240	0.810	0.928	0.967
Ours	`Flat`	0.141	1.270	5.779	0.239	0.814	0.927	0.966
Ours	`V+C+F`	0.139	1.211	5.702	0.239	0.816	0.928	0.966

E. Generalize to other datasets: Make3D

To showcase the generalizability of our approach, we follow the convention of [7], [31], [34], [37]: Our model trained *only* on KITTI Eigen split is directly tested on Make3D [26]. Make3D contains 534 images with 2272×1707 resolution, of which 134 are used for testing.[3] Low resolution ground truth depths are given as 305×55 range maps and must be resized and interpolated for evaluation. We follow [7] and [37] in applying a central cropping to generate a 852×1707 crop centered on the image. We use the standard $C1$ evaluation metrics for Make3D and measure our performance on depths less than 70 meters. Table V shows a quantitative comparison to the competitors, both supervised and self-supervised, with two different training settings. Note that the results of [11], [16], [18] are directly taken from [7]. Since the exact cropping scheme used in [7] is not available, we re-implemented it closely following the description in [7]. We trained our model on KITTI Eigen split and compared

[3]Ideally we want to test on the whole Make3D dataset since we do not train on Make3D, but other methods to which we compare train on it. For a fair comparison, we only use the 134 images for testing.

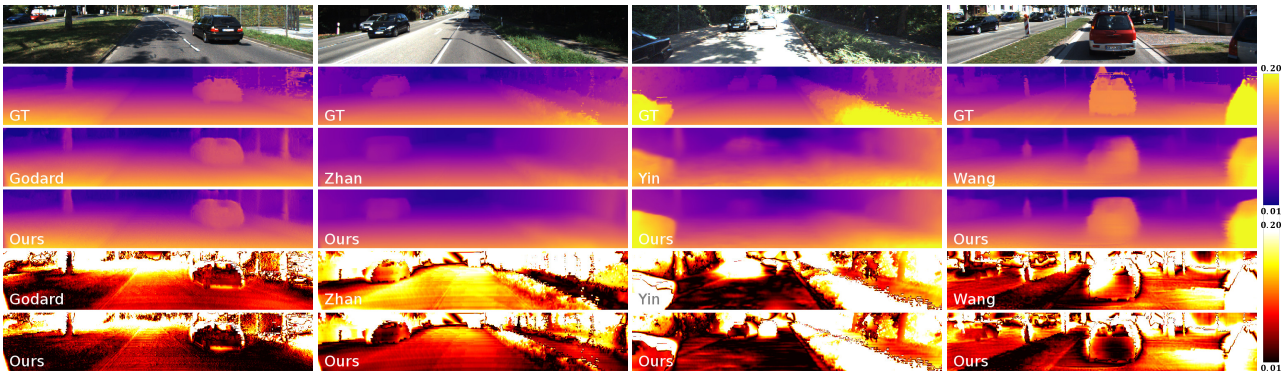

Fig. 2: *Qualitative results on KITTI Eigen split.* (best viewed at 5× with color) Top to bottom, each column shows an input RGB image, the corresponding ground truth inverse depth map, the predictions of baseline models trained without and with our priors, AbsRel error maps of baseline models trained without and with our priors. All the models are trained on KITTI Eigen split. For the purpose of visualization, ground truth is interpolated and all the images are cropped according to [5]. For the error map, darker means smaller error. Typical image regions where we do better (darker in the error map) include cars, roads and walls.

TABLE V: Generalizability test on Make3D.

Method	Supervision	AbsRel	SqRel	RMSE	$\log_{10}$
TrainSetMean	Depth	0.893	15.517	11.542	0.223
Karsch [11]	Depth	0.417	4.894	8.172	0.144
Liu [18]	Depth	0.462	6.625	9.972	0.161
Laina [16]	Depth	**0.198**	**1.665**	**5.461**	**0.082**
Godard [7] VGG	Stereo	0.468	9.236	12.525	0.165
Ours	Stereo	**0.458**	**8.681**	**12.335**	**0.164**
Zhou [37]	Mono	0.407	5.367	11.011	0.167
Yin [34]ResNet	Mono	0.376	4.645	10.350	0.152
Wang [31]PoseCNN+DDVO	Mono	0.387	4.720	**8.09**	0.204
Ours	Mono	**0.356**	**4.517**	10.047	**0.144**

against models of [7], [31], [34], [37] also trained on Eigen split (as provided by the authors) for a fair comparison.

A careful inspection of the baseline models (Godard [7] in stereo and Yin [34] in monocular supervision) versus ours reveals that the application of our losses does not hurt the generalizability of the baselines. Fig. 3 shows some qualitative results on Make3D. Though our model registers some failure cases in texture-less regions, a rough scene layout is present in the prediction. Regarding that the model is only trained on KITTI, of which the data modality is very different from that of Make3D, the prediction is sensible. But after all, a single image only affords to hypothesize depth, so we expect that any method using such predictions would have mechanisms to handle model deficiencies.

F. Evaluation on indoor datasets

To the best of our knowledge, none of the top-performing methods in self-supervised depth prediction have shown experimental results beyond planar motion, *i.e.,* driving scenarios such as KITTI and CityScapes, probably due to two reasons: Lack of rectified stereo pairs for training ([7], [35]) and difficulty to learn complex ego-motion along with depth prediction from video sequences ([31], [34], [37]).

However, with two modifications to the `GeoNet` model of Yin [34] – a multi-task learning approach where ego-motion and depth prediction are jointly learned, we managed to train our model and outperform `GeoNet` on publicly available

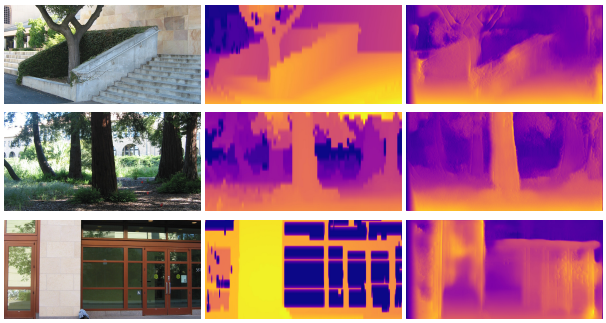

Fig. 3: *Qualitative results on Make3D.* Left to right, each row shows an input RGB image, the corresponding ground truth disparity map and our prediction. Our model is *only* trained on KITTI and directly applied to Make3D.

VISMA [4] dataset which features monocular videos of indoor scenes captured by a hand-held visual-inertial sensor platform under challenging motion. As a first modification, we replace the pose network in `GeoNet` with pose estimation from a VIO system [30], which makes the network easier to train (we call this model `OursVIO`). Second, to further improve the quality of predicted depth maps, we impose our gravity-induced regularization terms to `OursVIO`, where gravity is also estimated online by VIO. Our second model is named `OursVIO++`.

VISMA dataset contains time-stamped monocular videos (30 Hz) from a PointGrey camera and inertial measurements (100 Hz) from an Xsens unit, which are used in both VIO and network training. RGB-D reconstructions (dense point clouds) of the same scenes from a Kinect are also available, along with the spatial alignment $g_{\text{VIO}\leftarrow\text{RGBD}} \in \text{SE}(3)$ from RGB-D to VIO provided by the author. To get ground truth depth for cross-modality validation, we apply $g_{\text{VIO}\leftarrow\text{RGBD}}$ to the dense point clouds which are then projected to the PointGrey video frames. PSPNet trained on ADE20K [38] produces segmentation masks for training.[4] Of the $10K$ frames

[4]Among the 91 categories in ADE20K which PSPNet is trained on, we select "floor", "ceiling", "wall", "window", "door", "building", "chair", "cabinet", "desk", "table" to apply our losses.

TABLE VI: Quantitative results on VISMA validation.

Method	Error metric				Accuracy ($\delta <$)		
	AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
GeoNet	0.204	0.157	0.518	0.250	0.702	0.914	0.975
OursVIO	0.154	0.111	0.446	0.211	0.796	0.940	0.983
OursVIO++	**0.149**	**0.105**	**0.421**	**0.202**	**0.820**	**0.947**	0.983

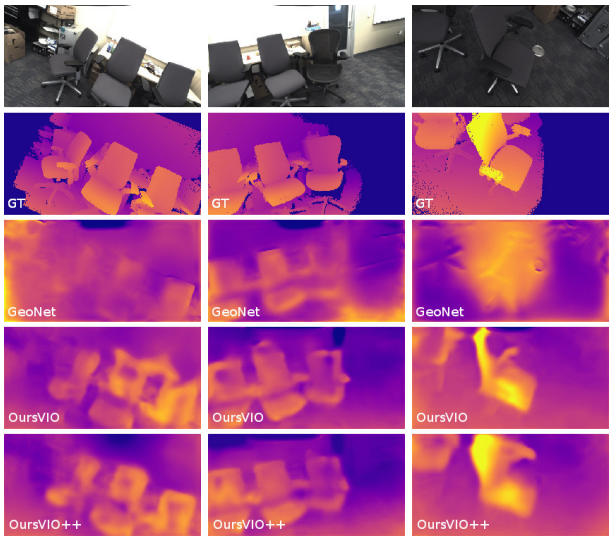

Fig. 4: *Qualitative comparison on VISMA validation.* Top to bottom, each column shows an input RGB image, the corresponding ground truth inverse depth map, results of GeoNet (baseline), OursVIO, and OursVIO++. Both OursVIO and OursVIO++ show largely improved results over the baseline, especially for images captured at extreme viewpoint (large in-plane rotation and top-down view). OursVIO++ (with gravity-induced priors) further improves over OursVIO (without priors) at planar regions, *e.g.,* the chair backs, where holes have been filled.

in VISMA, we remove static ones and construct 3-frame sequences (triplet) which are five frames apart in the original video to ensure sufficient parallax, resulting 8,511 triplets in total. We randomly sample 100 triplets for validation and use the rest for training. Fig. 4 and Table VI show comparisons of GeoNet, OursVIO and OursVIO++, all trained from scratch on VISMA until validation error stops decreasing. Both OursVIO and OursVIO++ improve over the baseline model by a large margin. Moreover, OursVIO++ trained with our gravity-induced losses has the capability to further refine results of OursVIO trained without our losses.

VI. Discussion

Gravity informs the shape of objects populating the scene, which is a powerful prior to visual scene analysis. We have presented a simple illustration of this power by adding a prior to standard monocular depth prediction methods that biases the normals of surfaces of known classes to align to gravity or its complement. Far more can be done: While in this work we use known biases in the shape of certain object classes, such as the fact that roads tend to be perpendicular to gravity, in the future we could learn such biases directly.

References

[1] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *ICML*, 2017.
[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.
[3] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014.
[4] X. Fei and S. Soatto, "Visual-Inertial Object Detection and Mapping," in *ECCV*, 2018.
[5] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *ECCV*. Springer, 2016.
[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *IJRR*, 2013.
[7] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.
[8] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3d scene reconstruction and class segmentation," in *CVPR*, 2013.
[9] K. He, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *CVPR*, 2016.
[10] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother, "Analyzing modular CNN architectures for joint depth prediction and semantic segmentation," in *ICRA*, 2017.
[11] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *ECCV*. Springer, 2012.
[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[13] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, "Learning-based, automatic 2d-to-3d image and video conversion," *IEEE Trans. on Image Proc.*, vol. 22, no. 9, pp. 3485–3496, 2013.
[14] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011.
[15] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *ECCV*. Springer, 2014.
[16] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016.
[17] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa, "Planenet: Piece-wise planar reconstruction from a single rgb image," in *CVPR*, 2018.
[18] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *PAMI*, vol. 38, no. 10, pp. 2024–2039, 2016.
[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
[20] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *CVPR*, 2018.
[21] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016.
[22] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *ICRA*, 2007.
[23] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "GeoNet: Geometric neural network for joint depth and surface normal estimation," in *CVPR*, 2018.
[24] C. Russell, P. Kohli, P. H. Torr, *et al.*, "Associative hierarchical crfs for object class image segmentation," in *ICCV*, 2009.
[25] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *NIPS*, 2006.
[26] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *PAMI*, vol. 31, no. 5, pp. 824–840, 2009.
[27] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008.
[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, 2012.
[29] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image," in *ICLR*, 2015.
[30] K. Tsotsos, A. Chiuso, and S. Soatto, "Robust inference for visual-inertial sensor fusion," in *ICRA*, 2015.
[31] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018.
[32] P. Wang, X. Shen, B. Russell, S. Cohen, B. Price, and A. L. Yuille, "Surge: Surface regularized geometry estimation from a single image," in *NIPS*, 2016.
[33] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in *ECCV*. Springer, 2016.
[34] Z. Yin and J. Shi, "Geonet: Unsupervised learning of deep depth, optical flow and camera pose," in *CVPR*, 2018.
[35] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *CVPR*, 2018.
[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.
[37] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.
[38] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, "Scene parsing through ade20k dataset"," in *CVPR*, 2017.