Geo-Supervised Visual Depth Prediction

Xiaohan Fei  Alex Wong  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, yet no artificial visual learning system explicitly exploits this bias. The shape of many artifacts, such as buildings and roads, and even natural objects such as trees, are influenced by gravity. Gravity provides a globally consistent orientation reference that can be reliably measured with cheap 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 choose the extreme example of predicting depth 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.

The ultimate test for a prior is whether it helps improve end-performance. To challenge our prior, we took two baselines, one for binocular (Sect. V-B) and one for monocular sequences (Sect. V-C), which were not the current top performers in the KITTI benchmark. We then added our priors, and tested the results against the top performers in the latest benchmark. Our prior made baseline methods beat top performers (Table II & III). We also performed ablation studies (Sect. V-D) and demonstrated our approach with visual-inertial systems on hand-held devices (Sect. V-F), which do not include ground truth.

II. RELATED WORK

Early learning-based depth prediction approaches [14], [16], [18], [32], [33] predict depth using local image patches which is then refined with Markov random fields (MRFs). Recent work [4], [5], [21] leverages deep networks to directly learn a representation for depth prediction where the networks are usually based on the multi-scale fully convolutional encoder-decoder structure. These methods are fully supervised and do not generalize well outside the datasets they are trained on. Latest self-supervised methods [8], [10], [46] have shown better performance on benchmarks (e.g., KITTI [9]), with better generalization.

There is a large body of work [26], [37], [42], [44] on self-supervised monocular depth prediction following Godard et al. [10] and Zhou et al. [46], 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 encoded into the network as additional loss terms for depth prediction [8], [27] and view synthesis [6], [40]. Our work is inline with these self-supervised approaches, but we also exploit class-specific regularizers beyond the generic ones.
In terms of exploiting the relation of different geometric quantities in an end-to-end learning framework, closely related work include [23], [30], [39], [41], where surface normals are explicitly computed by either a network [39] or some heuristics [30], [41] (Sect. III-B). While the former requires more computation, the latter relies on heuristics and thus is sub-optimal. In contrast, by using the proposed losses (Sect. III-A & III-B), we managed to directly regularize depth prediction via the depth-normal relation without all those modules to compute surface normals. In addition, both [39] and [23] are supervised, while ours is self-supervised with the photometric loss and guided by (i.e., conditioned on) global orientation and the semantic of the scene.

Earlier work on semantic segmentation [7], [34], [35] relied on hand-crafted features, and have been improved by incorporating global context using various structured prediction techniques [13], [19], [31]. Starting from the work of Long et al. [25], fully convolutional encoder-decoder networks [2], [22], [25], [45] have been a staple in semantic segmentation. Though we do not address semantic segmentation, we leverage on 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 [15], [38].

The idea of using class-specific priors to facilitate reconstruction is not new [11], [12], [20]. In [11], [12], class-specific shape priors in the form of spatially varying anisotropic smoothness terms are used in an energy minimization framework to obtain surface reconstruction of objects. The energy is then minimized via convex relaxation inspired by [43]. Though promising, this system does not scale well. An efficient inference framework [19] has been used with a CRF model over a voxel-grid to achieve real time performance by [20]. 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 in an optimization framework, while ours exploits semantics at an earlier stage – when generating such range images which themselves can serve as priors for the purpose of dense reconstruction and other inference tasks.

III. METHODOLOGY

In this section, we introduce SIGL – semantically informed geometric loss. At training time, SIGL serves as a regularizer which exploits the geometric relation between depth and gravity and is informed by semantic segmentation, or in other words, conditioned on object categories. Fig. [1] illustrates part of our training diagram. In Sect. III-C, we review two baseline models and show that the application of our loss 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 \( c(x, y) = \text{“road”} \) is often associated to a normal plane oriented along the vertical direction (direction of gravity), whereas \( c = \text{“building”} \) associated a normal direction on the horizontal plane. 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

\[
X = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} Z(x, y) \quad (1)
\]

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 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 \( X_i^T N = 1 \) for all \( i \), assuming the plane does not go through the optical center. Stacking all the points into matrix \( \bar{X} = [X_1, X_2, \ldots X_M]^T \), we have \( \bar{X} N = 1 \), a vector of \( M \) ones, where \( M = |\Omega| \) is the cardinality of the set \( \Omega \). 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)

\[
(I - \frac{1}{M} N^T) \bar{X} N = 0. \quad (2)
\]

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

\[
L_{HP}(\Omega_{HP}) = \frac{1}{|\Omega_{HP}|} \| (I - \frac{1}{|\Omega_{HP}|} N^T) \bar{X} ||_2^2 \quad (3)
\]

where \( N \) in Eq. (2) is replaced by normalized gravity \( \gamma \) due to the homogeneity of Eq. (3), and the squared norm is taken assuming the network predicts per-pixel depth \( Z(x, y) \) up to 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 \( \gamma \) 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 \( \gamma \), i.e., \( N \in \mathbb{N}(\gamma) \). Thus, the vertical plane loss reads

\[
L_{VP}(\Omega_{VP}) = \min_{N \in \mathbb{N}(\gamma)} \frac{1}{|\Omega_{VP}|} \| (I - \frac{1}{|\Omega_{VP}|} N^T) \bar{X} N ||_2^2 \quad (4)
\]

where the constraint \( \|N\| = 1 \) avoids trivial solutions \( N = 0 \) again due to the homogeneity of the objective; \( \Omega_{VP} \) 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.
The depth prediction training time, gravity extracted from inertial measurements is used to bias the network takes an RGB image as the only input and outputs a depth map. At training time, gravity extracted from inertial measurements is used to bias the depth prediction selectively. Note we do not predict semantic labels as extra outputs [15], [38] nor do we explicitly compute surface normals [37], [41]. Instead, semantic segmentation only provides a selection mechanism for us to apply our losses. In our experiments, we use a black-box PSP-Net to obtain segmentation. The other identical stream of the network and the photometric losses used for training are omitted in this figure.

B. Explanation of the objectives

Our idea is essentially to use priors about surface normals to regularize depth prediction. An intuitive way to achieve this is to first compute the surface normals from the depth values 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 estimation method involves normal estimation from depth, which can be problematic, especially with a simplistic but noisy normal

Let \( \mathbf{X} \) be the sample mean of the 3D coordinates and the horizontal plane loss \( L_{HP} \) reads

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

1For 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.

which is the sample variance of the 3D coordinates projected to the direction of gravity \( \gamma \) (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.

Similarly, to minimize \( L_{V P} \) 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 \( L_{V P} \), which explains the constrained quadratic problem in \( L_{V P} \). Though this can be solved via eigendecomposition, the gradients of the solver – needed in back propagation – 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 simply compute the minimum of the objective over the samples as an approximation to the loss.

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 – left-right consistency [10] and GeoNet [42] – and perform both quantitative and qualitative comparisons. The former is trained with rectified stereo image pairs while the latter uses monocular videos. At test time, both settings result in a system that takes a single image as input and predicts a depth/disparity map as output.

1) Training with stereo pairs: At training time, our first baseline model [10] takes a single left image as its input and predicts two disparity maps \( D_l, D_r : \mathbb{R}^2 \supset \Omega \rightarrow \mathbb{R}_+ \) with respect to 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 (see details in [10]). Here we address the view synthesis loss, which serves as the data term and is part of the appearance loss:

\[
L_{VS}^L = \frac{1}{|\Omega|} \sum_{(x,y) \in \Omega} \| I_l(x,y) - I_r^L(x + D_l(x,y),y) \|_1. \tag{8}
\]

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 \( I_r^L(x + D_l(x,y),y) \) according to the left disparity prediction \( D_l(x,y) \). The right view synthesis loss is constructed in the same way.

2) Training with monocular videos: To train our second baseline model [42], a single target 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:

\[
L_{VS} = \frac{1}{|W_t|} \sum_{(x,y) \in \Omega} \sum_{t' \in W_t} \| I_t(x,y) - I_t'(\pi(g_{t'}, \mathbf{X})) \|_1 \tag{9}
\]

which is the difference between the target view \( I_t \) and neighboring frames \( I_{t'} \) warped to the target view. \( \mathbf{X} \) is
the back-projected point defined in Eq. (1). $\pi$ is a central (perspective) projection, and $R_{bt}$ is the relative camera pose up to an unknown scale predicted by an auxiliary pose network which takes both $I_b$ 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. Detailed discussion about other losses serving as loss at training time; at test time, the depth network alone is needed to perform depth prediction with a single image as its input. Detailed discussion about other losses serving as 

IV. IMPLEMENTATION DETAILS

A. Semantic segmentation

At training time, we use PSP-Net [45] pre-trained on the CityScapes dataset [3] 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 PSP-Net, of which the most likely class $c(x, y) \in C$ determines the orientation of the surface where the back-projected point $X$ sits. We group the 19 classes into 7 categories$^{2}$ according to the CityScapes benchmark and test our losses on 4 of them, i.e., “flat”, “vehicle”, “construction” and “nature”. The other 3 categories do not have well-defined surface normals and thus are not tested.

B. Gravity

For imagery captured by a static platform equipped with an inertial measurement unit (IMU), one can use the gravity $g_b \in \mathbb{R}^3$ measured in the body frame (coinciding with the IMU frame) and simply apply the body-to-camera rotation $R_{db} \in SO(3)$ to obtain the gravity in the camera frame $g = R_{db}g_b$ which is then used in Eq. (3) and (4). For moving platforms, one resorts to robust visual-inertial odometry, which is well studied [28], [36]. In Sect. V-F, we demonstrated our approach on data captured by a Google Tango table in hand-held mode.

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 ($a_b \in \mathbb{R}^3$) and the spatial frame ($a_s \in \mathbb{R}^3$), we are able to compute the spatial-to-body rotation $R_{bs} \in SO(3)$ and then bring the gravity $g_s = [0, 0, 9.8]^{T}$ from the spatial frame to the camera frame $g = R_{sb}R_{bs}g_s$. In all settings, $R_{sb}$ (rotational part of the body-to-camera transformation) is obtained via offline calibration procedures.

C. Training details

Tensorflow and a GTX 1080 Ti GPU are used in our experiments. ADAM [17] optimizer with an initial learning rate of 0.0002 is used in both training settings. For the stereo training setting, each iteration with a mini-batch size of 8 256 x 512 RGB images takes ~280ms. For the monocular training setting, each mini-batch is composed of 4 3-frame (128 x 416 RGB) sequences and takes ~290ms per iteration.

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.

https://developers.google.com/ar/
consecutive frames are exploited in addition to rectified stereo pairs. Cap X meters means depth predictions are capped at X meters. Results of Zhan et al. [44] are taken from their paper. The rest of the results are taken from [10], unless otherwise stated.

We want to remind the reader that the baseline model on top of which we built ours is Godard et al. [10] which originally performs worse than Zhan et al. [44] by a large margin, but by applying our losses to Godard et al. [10] at training time we managed to boost its performance† and make it perform even better than Zhan et al. [44] at test time. Note Zhan et al. [44] also exploits temporal information in addition to stereo pairs for training while we do not. Fig. V-C shows a head-to-head qualitative comparison of ours and the baseline model.

C. Training with monocular videos

To demonstrate the effectiveness of our loss in the second training setting, we impose SIGL to our second baseline [42] which is a self-supervised depth network trained with monocular videos. Using the KITTI Eigen split, we follow the training and validation 3-frame sequence selection proposed by [46] where the first and third frames are treated as the source views and the central (second) frame is treated as the target as in Eq. (9). 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 [5]. The same training and evaluation scheme is also applied to other recent top-performing methods [26], [37], [46] in addition to the selected baseline [42].

Table III shows detailed comparisons against state-of-the-art self-supervised methods trained using monocular video sequences. All the models are trained only with monocular videos. Among many variants of Wang et al. [37] and Yin et al. [42], the top-performing models are shown in the table. Results of [37] capped at 50 meters are not available at the time of submission. Unless otherwise stated, all the results are taken from each author’s paper respectively. Notations have the same meaning as in Table II. Compared to [42] which our model is based on, we have reduced the prediction error by a large margin.† Moreover, by adding the proposed losses we even outperform [37], which originally performed better than our baseline model [42]. For qualitative results, see Fig. V-C which shows typical image regions where we do better.

D. Ablation study

To study the contribution of each individual semantic category to the performance improvement in terms of visual depth prediction, we performed an ablation study: We selectively 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. [10] is the baseline model where only the most generic regularizers, e.g., smoothness and consistency, are used. Second column indicates the semantic category of 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. This is expected because most of the KITTI images contain a large portion of roads and/or sidewalks. What can also be observed is that regularization of the “construction” and “vehicle” category provides reasonable improvements while the “nature” category helps a little.

E. Generalize to other datasets: Make3D

To showcase the generalizability of our approach, we follow the convention of [10], [37], [42], [46]: Our model trained only on KITTI Eigen split is directly tested on Make3d [33]. Make3d contains 534 images with 2272 × 1707 resolution, of which 134 are used for testing. Low resolution ground truth depths are given as 305 × 55 range maps and must be resized and interpolated for evaluation. We follow

---

**Table II**

| Method | Sup | Error metric | Accuracy (δ < 1) | Depth: cap 80m |
|-------|-----|--------------|-----------------|----------------|
|       |     | AbsRel | SqRel | RMSE | RMSE log |佐 | 佐 |
| TrainSetMean | D | 0.361 | 4.826 | 8.102 | 0.377 | 0.638 | 0.804 | 0.894 |
| Eigen | D | 0.214 | 1.605 | 6.563 | 0.292 | 0.673 | 0.884 | 0.957 |
| Eigen | D | 0.203 | 1.548 | 6.307 | 0.282 | 0.702 | 0.890 | 0.958 |
| Liu | D | 0.201 | 1.584 | 6.271 | 0.273 | 0.680 | 0.898 | 0.967 |
| Godard [10] | S | 0.148 | 1.344 | 5.927 | 0.247 | 0.803 | 0.922 | 0.964 |
| Zhan [44] | S+T | 0.144 | 1.391 | 5.869 | 0.241 | 0.803 | 0.928 | 0.969 |
| Ours | S | 0.139 | 1.211 | 5.702 | 0.239 | 0.816 | 0.928 | 0.966 |

| Method | Sup | Error metric | Accuracy (δ < 1) | Depth: cap 80m |
|-------|-----|--------------|-----------------|----------------|
|       |     | AbsRel | SqRel | RMSE | RMSE log |佐 | 佐 |
| Zhou [46] | S | 0.183 | 1.595 | 6.709 | 0.270 | 0.734 | 0.902 | 0.959 |
| Mahjourian [26] | S | 0.163 | 1.240 | 6.220 | 0.250 | 0.762 | 0.916 | 0.968 |
| Yin [42] | S+T | 0.155 | 1.296 | 5.857 | 0.233 | 0.793 | 0.931 | 0.973 |
| Wang [37] | S+T | 0.151 | 1.257 | 5.583 | 0.228 | 0.810 | 0.936 | 0.974 |
| Ours | S+T | 0.142 | 1.124 | 5.611 | 0.223 | 0.813 | 0.938 | 0.975 |

**Table III**

| Method | Sup | Error metric | Accuracy (δ < 1) | Depth: cap 80m |
|-------|-----|--------------|-----------------|----------------|
|       |     | AbsRel | SqRel | RMSE | RMSE log |佐 | 佐 |
| Zhou [46] | S | 0.201 | 1.391 | 5.181 | 0.284 | 0.698 | 0.900 | 0.964 |
| Mahjourian [26] | S | 0.155 | 0.927 | 4.549 | 0.231 | 0.781 | 0.931 | 0.975 |
| Yin [42] | S+T | 0.147 | 0.936 | 4.348 | 0.218 | 0.810 | 0.941 | 0.977 |
| Ours | S+T | 0.135 | 0.834 | 4.193 | 0.208 | 0.831 | 0.948 | 0.979 |

Model variants: 1Latest 2ResNet 3PoseCNN+DDVO 4: baseline model which ours is built atop; 5: state of the art.
[10] and [46] 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 IV shows a quantitative comparison to the competitors, both supervised and self-supervised, with two different training settings. Note that the results of [16], [21], [24] are directly taken from [10], and [37] from the corresponding paper, of which the central crop used in their evaluation is not provided. However, we managed to evaluate [10], [42], [46], whose code is publicly available, with our own cropping scheme. A careful inspection of the baseline models ( [10] in stereo and [42] in monocular supervision) versus ours reveals that the application of our loss does not hurt the generalizability of the baselines. Fig. 3 shows some qualitative results on Make3D. Though our model is only trained on KITTI, it is able to capture the rough layout of the scene.

### F. Demonstration on hand-held VIO dataset: PennCOSYVIO

In addition to KITTI, which focuses on the driving scenario and features planar motion, we also test our system on PennCOSYVIO [29] dataset, which has both visual and inertial measurements captured by a hand-held sensor-rich rig. Among the various sensory data, we use the RGB video and inertial measurements streamed from the Google Tango tablet mounted on the rig in our experiment. However, since the dataset is originally built for benchmarking visual-inertial odometry, it does not have enough data (several minutes of RGB video streams in total) to train a deep neural network from scratch. To this end, we first train our model on monocular videos from the KITTI dataset and then fine-tune it on the PennCOSYVIO dataset. The direction of gravity is first estimated with some static frames at the beginning of the video and then propagated forward via the pose estimated by the onboard visual-inertial odometry algorithm.

Quantitative evaluation is not available due to the lack of

---

#### TABLE IV

Ablation study on KITTI Eigen Split.

| Method   | Category | Error metric   | Accuracy (δ <) | AbsRel | SqRel | RMSE | log
|----------|----------|----------------|----------------|--------|-------|------|-----
| Ours     | N        | AbsRel, SqRel  | 0.384          | 0.247  | 0.239 | 0.816 | 0.928
| Ours     | V        | AbsRel, SqRel  | 0.384          | 0.247  | 0.239 | 0.816 | 0.928
| Ours     | C        | AbsRel, SqRel  | 0.384          | 0.247  | 0.239 | 0.816 | 0.928
| Ours     | F        | AbsRel, SqRel  | 0.384          | 0.247  | 0.239 | 0.816 | 0.928
| Ours     | F+C+V    | AbsRel, SqRel  | 0.384          | 0.247  | 0.239 | 0.816 | 0.928

N=Nature, V=Vehicle, C=Construction, F=Flat

---

#### TABLE V

Generalizability test on Make3D.

| Method   | Supervision | AbsRel | SqRel | RMSE | log
|----------|-------------|--------|-------|------|-----
| Godard   | Stereo      | 0.468  | 9.236 | 12.525 | 0.204 |
| Ours     | Stereo      | 0.458  | 8.681 | 12.335 | 0.164 |
| Zhou     | Mono        | 0.407  | 5.367 | 11.011 | 0.167 |
| Yin      | Mono        | 0.376  | 4.645 | 10.350 | 0.152 |
| Wang     | Mono        | 0.387  | 4.720 | 8.09   | 0.204 |
| Ours     | Mono        | 0.356  | 4.517 | 10.047 | 0.144 |

Model variants: ResNet, PoseCNN+DDVO

---

Fig. 2. Qualitative results on KITTI Eigen split. (best viewed at 5× with color) Top to bottom: Input RGB image, ground truth disparity map, baseline (Yin et al. [42] and Godard et al. [10]), ours, AbsRel error map of baseline models, AbsRel error map of ours. For the baseline model, left two columns correspond to [42] and right two columns correspond to [10]. For the purpose of visualization, ground truth is interpolated and cropped [8]. 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.

Fig. 3. Qualitative illustration on Make3D. From left to right, the input RGB image, ground truth disparity and our prediction. Our model is only trained on KITTI Eigen split and directly applied to Make3D.
of ground truth depth map or point cloud. Nevertheless, illustrative results can be found in Fig. 4, where a rough scene layout is present in our prediction, though small details including poles, cars and people far away are not captured well. These failure cases are somewhat caused by the small amount of training data, but after all, a single image only affords hypothesizing depth, so we expect that any method using such predictions would have mechanisms to handle those.

VI. DISCUSSION

Gravity informs the shape of objects populating the scene, which is a powerful prior for 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 to 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 International Conference on Machine Learning, 2017, pp. 136–145.
[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
[3] 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 Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[4] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in Proc. IEEE International Conference on Computer Vision, 2015, pp. 2650–2658.
[5] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Advances in Neural Information Processing Systems, 2014, pp. 2366–2374.
[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, “DeepStereo: Learning to predict new views from the world’s imagery,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[7] B. Fulkerson, A. Vedaldi, and S. Soatto, “Class segmentation and object localization with superpixel neighborhoods,” in Proc. IEEE International Conference on Computer Vision, 2009, pp. 670–677.
[8] R. Garg, V. K. BG, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in Proc. European Conference on Computer Vision, Springer, 2016.
[9] A. Geiger, P.Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research, 2013.
[10] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[11] C. Hane, N. Savinov, and M. Pollefeys, “Class specific 3d object shape priors using surface normals,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 652–659.
[12] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, “Joint 3d scene reconstruction and class segmentation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 97–104.
[13] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, “Multiscale conditional random fields for image labeling,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[14] D. Hoiem, A. A. Efros, and M. Hebert, “Recovering surface layout from an image,” International Journal of Computer Vision, vol. 75, no. 1, pp. 151–172, 2007.
[15] 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 Proc. IEEE International Conference on Robotics and Automation, 2017, pp. 4620–4627.
[16] K. Karsch, C. Liu, and S. B. Kang, “Depth extraction from video using non-parametric sampling,” in Proc. European Conference on Computer Vision. Springer, 2012, pp. 775–788.
[17] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. International Conference on Learning Representation, 2015.
[18] J. Konrad, M. Wang, F. Ishwar, C. Wu, and D. Makherjee, “Learning-based, automatic 2d-to-3d image and video conversion,” IEEE Transactions on Image Processing, vol. 22, no. 9, pp. 3485–3496, 2013.
[19] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in Advances in Neural Information Processing Systems, 2011, pp. 109–117.
[20] A. Kundu, Y. Li, F. Dellerta, F. Li, and J. M. Rehg, “Joint semantic segmentation and 3d reconstruction from monocular video,” in Proc. European Conference on Computer Vision, Springer, 2014, pp. 703–718.
[21] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in Proc. IEEE International Conference on 3D Vision, 2016, pp. 239–248.
[22] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” in Proc. International Conference on Learning Representations, 2015.
[23] I. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa, “Planenet: Piece-wise planar reconstruction from a single rgb image,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2579–2588.
[24] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 10, pp. 2024–2039, 2016.
[25] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[26] R. Majhouri, M. Wicke, and A. Angelova, “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[27] N. Mayer, E. Ilg, P. Hauser, 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 Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[28] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in Proc. IEEE International Conference on Robotics and Automation, 2007, pp. 3565–3572.

[29] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, “Pencosyvio: A challenging visual inertial odometry benchmark,” in Proc. IEEE International Conference on Robotics and Automation, 2017, pp. 3847–3854.

[30] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, “GeoNet: Geometric neural network for joint depth and surface normal estimation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[31] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, “Pencozyvio: A challenging visual inertial odometry benchmark,” in Proc. IEEE International Conference on Robotics and Automation, 2017, pp. 3847–3854.

[32] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in Advances in Neural Information Processing Systems, 2006, pp. 1161–1168.

[33] A. Saxena, M. Sun, and A. Y. Ng, “Make3d: Learning 3d scene structure from a single still image,” IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 5, pp. 824–840, 2009.

[34] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[35] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” International Journal of Computer Vision, vol. 81, no. 1, pp. 2–23, 2009.

[36] K. Tsotsos, A. Chiasso, and S. Soatto, “Robust inference for visual-inertial sensor fusion,” in Proc. IEEE International Conference on Robotics and Automation, 2015.

[37] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, “Learning depth from monocular videos using direct methods,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[38] P. Wang, X. Shen, Z. Liu, S. Cohen, B. Price, and A. L. Yuille, “Towards unified depth and semantic prediction from a single image,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2800–2809.

[39] P. Wang, X. Shen, B. Russell, S. Cohen, B. Price, and A. L. Yuille, “Surge: Surface regularized geometry estimation from a single image,” in Advances in Neural Information Processing Systems, 2016, pp. 172–180.

[40] J. Xie, R. Girshick, and A. Farhadi, “Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks,” in Proc. European Conference on Computer Vision. Springer, 2016.

[41] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, “Unsupervised learning of geometry with edge-aware depth-normal consistency,” arXiv preprint arXiv:1711.03665, 2017.

[42] Z. Yin and J. Shi, “Geonet: Unsupervised learning of deep depth, optical flow and camera pose,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[43] C. Zach, C. Häne, and M. Pollefeys, “What is optimized in tight convex relaxations for multi-label problems?” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1664–1671.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.

[45] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.