Improving depth estimation using map-based depth priors

Vaishakh Patil[1], Alexander Liniger[1], Dengxin Dai[1,2] and Luc Van Gool[1,3]

***Abstract*—Accurate scene depth is fundamental for robot scene understanding as it adds spatial reasoning. However, accurate scene depth often comes at the cost of expensive additional depth sensors. In this article, we propose to use map-based depth data as an additional input instead of expensive depth sensors. Such an approach is especially appealing in autonomous driving since map-based depth is commonly available from high-definition maps. To validate this approach, we propose a mapping method that works with common autonomous driving datasets and allows for precise localization using a mix of GNSS-INS and image-based techniques. Furthermore, we propose an entirely learnable three-stage network that handles foreground-background mismatches between the map-based prior depth and the actual scene. Finally, we validate the performance of our method in comparison to several baseline methods and SOTA depth completion methods receiving map-based depth as an input. Our method significantly outperforms these methods both in quantitative and qualitative results. Moreover, our method achieves better metric-scale predictions compared to image-only approaches.**

***Index Terms*—Deep Learning for Visual Perception, RGB-D Perception, Sensor Fusion, Novel Deep Learning Methods, Autonomous Vehicle Navigation,**

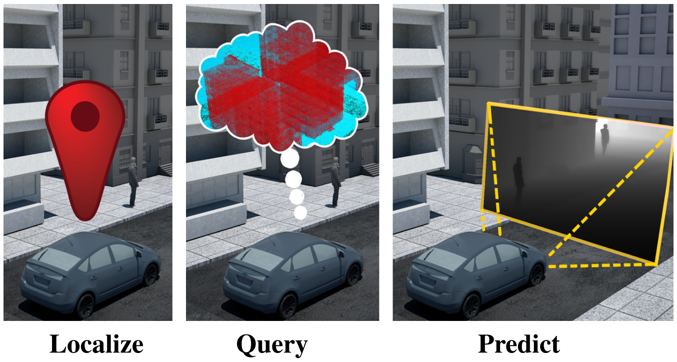

Fig. 1. Autonomous vehicles operate in an outdoor environment and are often equipped with GNSS sensors and HD maps. Given the ability of such a robotic platform to localize itself, it can query the HD map for a depth prior. This prior depth information can then be fused with image information to generate accurate depth maps of the scene.

I. INTRODUCTION

ACCURATE scene depth understanding is essential for a variety of robotic applications such as path planning, augmented reality, and autonomous driving. These robots have to rely on engineered depth estimation systems as opposed to humans' depth assessment ability through their visual cortex. Historically, robotic applications perceived depth measurements through stereo cameras or active depth sensors like LiDARs, Time-of-Flight (ToF) sensors, or structured light cameras. Although these systems can generate reliable depth maps using well-defined mathematical principles, they are more expensive than RGB cameras with short to medium operating range (<100m) and can have holes in depth maps. In case of the popular mechanical spinning LiDARs, integration is complex due to its bulkiness and rotating mechanism, and operation is restricted due to its fragility under adverse weather conditions or vibrations. ToF cameras are often short-range and cause erroneous depth measurement in multi-camera setups [1]. Finally, estimating depth from a single camera is an ill-posed problem since a single view of the scene can be interpreted as a large number of different 3D scenes generating ambiguous depth information. However, recent advancements in deep learning have allowed for a massive leap in this area [2], bringing practicality.

Motivation: Several robotic applications use depth information to interact with the surrounding environment. This demands that the estimated depth maps are causal and obtained with minimal latency and high throughput. One of the feasible solutions considering these objectives is to use prior data. Previously, several methods in different robotic applications have relied upon prior knowledge of the scene [3]–[5]. Likewise, we assert that depth estimation can benefit from prior depth information, specifically from map-based depth information. Moreover, various robotic applications require performing repetitive tasks in the same environment mostly consisting of large static regions. It is also possible to capture and accumulate the 3D map of these static scene in advance. Hence, mapping is an intuitive approach used in robotics for tasks such as localization as the stored information can be exploited during subsequent explorations. Similarly, depth estimation methods can benefit from such map based prior depth information to generate scale-aware, high quality depth maps.

Application: Let us consider the case of autonomous vehicles (AV). Traditionally, human drivers are known to use maps to guide themselves to a location. Similar AVs also use maps

Manuscript received: September, 09, 2021; Revised December, 5, 2021; Accepted January, 3, 2022.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments.

This work is funded by Toyota Motor Europe via the research project TRACE Zurich.

[1] Vaishakh Patil, Alexander Liniger, Dengxin Dai and Luc Van Gool are with the Computer Vision Lab, ETH Zurich, 8092 Zurich, Switzerland {patil, alex.liniger, dai, vangool}@vision.ee.ethz.ch

[2] Dengxin Dai is with the MPI for Informatics, Saarbrucken, Germany

[3] Luc Van Gool is with the Toyota TRACE-Leuven at the Dept. of Electrical Engineering ESAT, KU Leuven, 3001 Leuven, Belgium

Digital Object Identifier (DOI): see top of this page.

but extend them with additional information which can assist AVs; these maps are called High-Definition(HD) maps. HD maps primarily contain detailed information of road elements (road shape, marking, barriers, traffic signs) and the point cloud of areas around the road. If the vehicle can localize itself accurately in these maps, the point clouds can provide the 3D context of the static scene over a large range. In this work, we argue that this information can be used to enhance the functionality of a monocular camera to estimate depth. Given a precise visual and GNSS-based localization [6], [7] we can use the 3D point cloud map information from this view to help the depth estimation of the current image (Fig. 1).

Contribution: In this article, we propose to enhance the traditional depth estimation from a monocular camera with the help of a map-based prior depth. This problem is tackled with a novel deep learning pipeline that fuses these previous depth measurements of nearby scenes with the current depth estimate. Our system is designed to handle dynamic objects in the scene and misalignment in localization in an automated, learnable fashion. Additionally, we show the advantages of our method by comparing it to the SOTA method in the depth completion realm, in the case only map-based depth is available. Finally, we present an extensive quantitative and qualitative evaluation of the proposed method along with a thorough ablation study.

II. Related Work

The field of depth estimation has been filled with a plethora of methods. Each of these methods is designed for the specific application of depth estimation. Given the breadth of this area, we outline the methods designed for monocular images relevant to our setup.

A. Learning depth from single image

The ambiguous task of learning depth from a single image can be approached by learning. Here we will outline the works related to predicting depth from a single RGB image. Saxena *et al.* [8] were one of the earliest to popularize this idea. Since then, several works have proposed to learn depth with different settings: Garg *et al.* [9], and Godard *et al.* [10] used images pairs, [11] used multiple overlapping images captured from different viewpoints, [12] utilizes binaural sounds. [13] utilized generative models and [14] used continuous 3D loss function to learn the depth. For this setup, deep learning-based methods were first proposed in [15]–[18], which use supervision obtained from active sensors (ToF, LiDAR sensors). Furthermore, methods were developed to learn directly from stereo pairs [10] combined with direct supervision from LiDAR sensors [19]. To avoid the dependency on expensive Ground Truth (GT) depth, self-supervised methods became popular, which use view synthesis or its variants as the supervisory signal [20]–[24]. Despite the popularity of these methods, they are often unstable to train, require hyperparameter tuning and suffer from scale ambiguity.

B. Depth completion methods

Alongside the advancements in single image depth estimation, it has been shown that additional depth information can significantly improve depth performance. One such task is depth completion, where the goal is to enhance sparse depth data (i.e., from a LiDAR sensor) with image guidance. The problem was first defined by the KITTI Depth Completion Benchmark [25]. Since then, numerous methods have been proposed, significantly increasing the quality of the results [26]–[32]. This line of work has been further extended for video data [33], sparse radar sensors [34], laser range finder [35], [36]. Other works have shown that sparse depth inputs from either SLAM or structure-from-motion systems [37], [38] can improve depth predictions.

There are classical methods that make use of prior depth information [39]–[42]. In contrast to our approach, these methods work by estimating the dense optical flow or 3D motion between the current scene and the corresponding previous scene. This information is used to warp the previous depth information to the current scene. [40], [42] estimate depth between two frames only by warping without accounting for the changes. These methods cannot be applied to dynamic scenes as they only account for small changes in depth or in-plane motion. [39] warps the depth image from an RGB-D image database to the current image by measuring optical flow between both RGB images. The warped depths are then optimized to estimate the depth of the current image. [41] first computes the rigid regions between consecutive images with optical flow. The new depth values are assigned from previous depth maps by using photometric error between the current and re-projected previous image, the estimated rigid motion and the previous depth map.

Overall, accurate sparse depth measurements can significantly boost the performance of image-based depth estimation. In our work, we aid RGB images with map-based prior depth leading to a similar conclusion.

III. Depth estimation using map-based depth priors

As discussed in the Introduction I, depth estimation from only a single image is ill-posed and suffers from several issues such as scale consistency. One approach to solve these issues is by fusing visual information with direct depth measurements by a secondary sensor. However, adding such a sensor adds a significant cost and increases energy consumption. In this work, we exploit the fact that robots often operate in a predefined limited operation space, i.e., geo-restricted autonomous cars. Thus, we propose to replace expensive depth sensors with map-based depth priors accessible as part of HD maps. Note that we still need expensive sensors for the HD map generation and supervision, but no additional sensors are required during operation.

Designing such a system comes with two main challenges; first, a HD map that stores depth data in a suitable format. Second, a robust depth fusion network to deal with the inconsistencies of the map depth data (foreground objects) and the potentially sparse supervision.

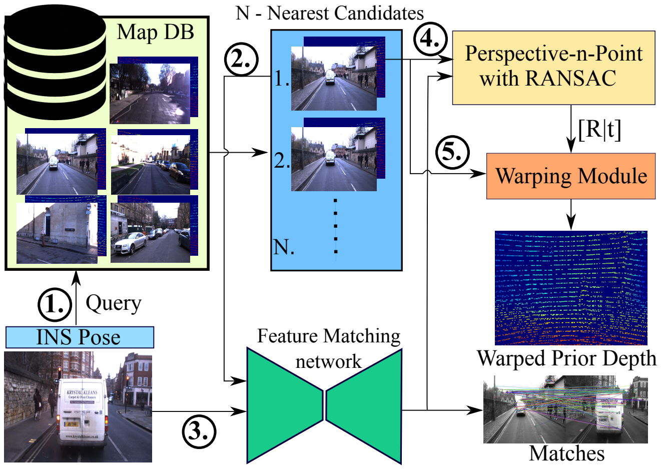

Fig. 2. Map query: Prior depth information is obtained through a query in the map database followed by Perspective-n-Point based warping. The steps of the pipeline are denoted in encircled bold numbers.

A. *Depth prior map*

In the context of autonomous driving, HD maps have become a crucial building block. Modern HD maps for autonomous driving have several components, from lane graph information to 3D point cloud data of the surrounding. We argue that this 3D data can also be used to improve depth estimation. Even though 3D point cloud based HD maps are commonly used for autonomous driving, they are rarely publicly available. Thus, in this paper, we develop a simple GNSS-INS referenced depth map, which can be applied to public datasets that sufficiently cover a geo-restricted area.

There is no fixed standard for how a 3D data HD-map should be structured, VarCity [43] for example, build a large scale 3D-City model, and TomTom's RoadDNA system representing the 3D lateral and longitudinal view of the roadway. However, we determined three main challenges when developing such a large-scale map.

Constructing a large-scale point cloud is complex. It requires the accumulation and processing of multiple point clouds captured by a depth sensor. Moreover, the size of the map generated by naively accumulating individual point clouds can get prohibitively large. Thus, efficient data management is needed.

Updating the existing point cloud can be as challenging as constructing it. The map should be set up such that new data improves the map and replaces outdated information, while the size of the resulting map should remain roughly constant.

Localization within the point cloud should be fast and precise. This is challenging due to the unordered nature of the point cloud. It can be addressed with additional metadata such as 3D point descriptors, GNSS referencing, and vision-based appearance information.

In this work, we propose a simple map for depth data that works with existing autonomous driving datasets. Therefore, instead of a global accumulation, we perform a temporal and spatial local accumulation of the LiDAR point cloud around each keyframe of the dataset. Thus, each local point cloud is registered to the ECEF (Earth-centered, Earth-Fixed) reference system. Since we can only rely on visual localization (no LiDAR at test time), we store the front-facing image along with the local point cloud. Thus, our HD-map is a list of locally accumulated 3D point clouds D_{map}, projected and clipped within the camera's field of view to reduce the size requirements. Additionally, we store the front-facing image I_{map} and the GNSS-INS location Loc_{map} =(lat_{map}, $long_{map}$), i.e. latitude and longitude information. This map setup allows for a straightforward generation since the point cloud accumulation is only done locally, avoiding size and global alignment issues. Updating the map is trivial as new elements can be simply added to the list, and old locally close entries can be deleted. Finally, for localization, we can use GNSS and image-based tools.

We acknowledge that our method is still rather simple for map-based vehicle localization, and it may not represent the state of the arts for localization. However, the focus of this work is to show that our depth estimation method is able to benefit from having prior map data and can do so even without using the most sophisticated localization methods. Developing and using more sophisticated localization methods is orthogonal to this work.

Map query: Localization within the map and retrieving the prior depth is fundamental for our method. Our approach uses a mix of GNSS and vision-based localization, and the full approach is shown in Fig. 2 and summarized in Algo. 1. First, we use the current GNSS location Loc_{curr}=(lat_{curr}, $long_{curr}$) to retrieve the n closest elements in the map in terms of the euclidean distance. Next, we use an image-based localization approach to find the relative transformation between the retrieved map elements image I_{map} and the current image I_{curr}. Following the recent advances in deep learning-based feature matching, we employ such methods (SuperGlue [44] in our case) to find robust correspondences between I_{curr} and I_{map}. Given the 2D correspondences and the corresponding 3D points given by D_{map}, we can estimate a relative pose $T_{map \rightarrow curr}$ using the Perspective-n-Point (PnP) algorithm. Now, using the transformation and the camera matrix K, we can warp the map depth to the current frame as,

$$\hat{D}_{curr} = K T_{map \rightarrow curr} D_{map} K^{-1} p_{map} \,, \tag{1}$$

where $p_{map} = (u, v, 1)$ are homogeneous image coordinates, and a hat indicates a warped map-based prior depth. Finally, we aggregate the n warped prior depth images, excluding views with less than τ points in the warped prior depth. For the remaining sections we do drop the $curr$ subscript for readability.

B. *Depth fusion network*

The goal of our depth fusion network is to use the warped prior depth $\hat{D}$ to improve the depth estimation from an image I. Thus, our fusion problem is similar to the depth completion problem but with different key challenges. From a high-level perspective, we have an RGB image $I(u, v)$ where (u, v) indicates the pixel location, as well as a sparse depth input $\hat{D}(u, v)$. Finally, we have the ground truth depth $D_{gt}(u, v)$,

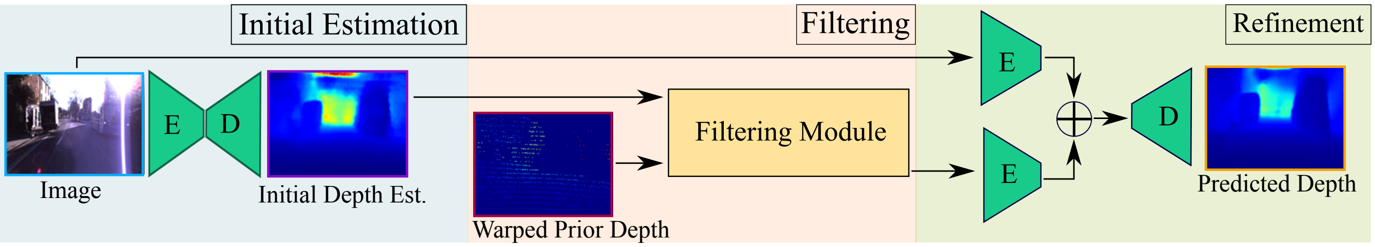

Fig. 3. The complete pipeline consist of a "Three Stage Prior Depth Fusion Network". The stages are labeled at the right-top corner of each block. The notations are as follows, $\bigoplus$: concatenation, E: Encoder, D: Decoder.

Algorithm 1 Get Depth Priors

```
function GenerateDepthPriors($\mathbf{N}$, $\tau$)
    $\mathbf{Lcurr} \leftarrow$ Current GNSS location.
    $\mathbf{L_{map}} \leftarrow$ List of all Map locations.
    $\mathbf{D^{idx}}, \mathbf{L^{idx}}, \mathbf{pt} \leftarrow \{\}$
    $\mathbf{idx} \leftarrow \textsc{Sort}(\operatorname{argmin} \|\mathbf{L_{curr}} - \mathbf{L_{map}}\|_2)$
    for $\textsc{TopElements}(\mathbf{idx}, \mathbf{N}) \rightarrow \mathbf{L_{map}}$ do
        $\mathbf{T} \leftarrow \textsc{PerspectiveNPoint}(\mathbf{I_{curr}}, \mathbf{I^{idx}_{map}}, \mathbf{D^{idx}_{map}})$
        $\mathbf{D_{warped}} \leftarrow \textsc{WarpDepth}(\mathbf{D_{map}}, \mathbf{T})$
        if no. of points in $\mathbf{D_{warped}} \geq \tau$ then
            $\mathbf{L^{idx}} \leftarrow \mathbf{L_{map}}$
            $\mathbf{D^{idx}} \leftarrow \mathbf{D_w}$
    for each $\mathbf{id}$ in $\mathbf{L^{idx}}$ do
        $\mathbf{pt} \leftarrow \textsc{Project}(\mathbf{D^{idx}_{id}}, \mathbf{K^{-1}})$
    $\hat{\mathbf{D}}_{\mathbf{curr}} \leftarrow \textsc{BackProject}(\mathbf{pt}, \mathbf{K})$
    return $\hat{\mathbf{D}}_{\mathbf{curr}}$
```

which is sparse but sufficiently dense to give meaningful supervision for the network. Given these ingredients, the goal is to learn a deep neural network $f_\theta(I, \hat{D})$ that maps the RGB image and warped depth image to a dense depth image $\bar{D}(u, v)$. In our setting all I, $\hat{D}$ and $\bar{D}$ have all the same resolution $H \times W$ and $\bar{D} > 0$. We train our neural network in a supervised fashion,

$$\min_{\theta} \mathcal{L}_M(D_{gt}, f_\theta(I, \bar{D})) , \tag{2}$$

where the loss $\mathcal{L}_M$ is a masked loss, which considers the binary mask $M(u, v)$ of the GT depth.

Different from the standard depth completion task, our depth estimation with map priors has several additional challenges. First, the depth prior is potentially wrong: the depth prior does not contain the current foreground objects, potentially contains foreground objects not present in the current scene, and small objects may be misaligned. Thus, the network needs to learn where to trust the map-based depth prior and where not. Second, given that the GT depth is sparse and the input prior gives incorrect cues, supervision becomes challenging. Thus, we need to include additional supervision signals allowing us to predict a high-quality dense depth. In the next section, we will discuss how we tackled these two challenges.

C. Three stage depth fusion network

To deal with issues of the map-based prior depth input, we propose a three-stage network (Fig. 3) consisting of an initial monocular estimation stage, a second filtering stage that adds the depth prior, and a final refinement stage.

Initial estimation: The initial estimation stage is a monocular depth estimation method that predicts a depth map $\bar{D}_{init}$ using the current image and is directly supervised by the GT depth. This initial estimate gives an anchor for what depth information can be extracted from the image. This anchor can be used in the second stage to decide where to trust the map-based depth prior.

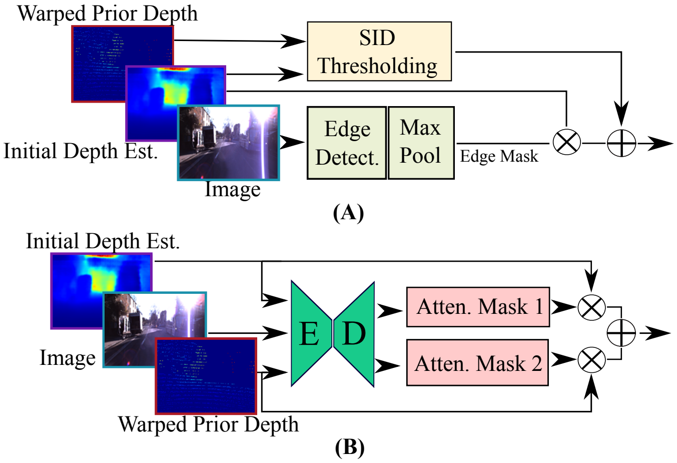

Fig. 4. The warped map depth often comes with several issues. To handle this, we propose two possible filtering modules: (A) Classical Filtering, (B) End-to-End Filtering. The output of this stage is forwarded to the refinement module as shown in Fig. 3. The notations are as follows, $\bigoplus$: concatenation, $\bigotimes$: element-wise multiplication, E: Encoder, D: Decoder.

Filtering stage: To deal with the issues in the map-based depth prior, we propose to use a filtering stage. We design two possible approaches for this stage (Fig. 4); both have the same goal, combining the depth map from the initial stage and the map-based prior depth. Therefore, two masks have to be estimated (Fig. 5), one for each of the depth maps. The first approach is a baseline method that is based on **classical** ideas, which uses SID filtering [45] to determine where to use the map-based depth prior, given the two depth maps as an input. The SID filter is used to compute a threshold to eliminate points from the prior depth map. More precisely, we compute the threshold as

$$\delta = \exp\left(\frac{d\log(\frac{\beta}{\alpha})}{K} + \log(\alpha)\right) , \tag{3}$$

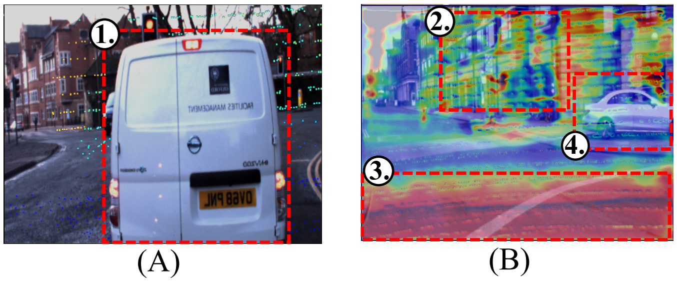

Fig. 5. The figure shows the following overlayed on the image: (A), filtered mask using classical filtering, and (B), generated mask using end-to-end filtering. In (1) and (4), the warped prior depth points projected on dynamic objects have been discarded. In (2) and (3), the warped prior depth points are retained and the initial predicted depth is given a lower weight.

where d is the prior depth at a specific pixel, and α and β are hyperparameters. Given this threshold, we remove points from the prior depth map if $|D_{map} - \bar{D}_{init}| \geq \delta$. This approach allows deciding where to trust the prior depth given the initial stage depth anchor. However, the initial stage depth $\bar{D}_{init}$ is not perfect as it often fails at edges in the image. Thus, we generate an edge map from the image, and we further process the edge map with a 3×3 max pool filter to make the edge regions more homogeneous and reduce noise. Finally, we threshold the edge map to generate a mask for the initial stage depth. The two masked depth maps are then concatenated and further processed by the refinement stage.

The second filtering approach is formulated as an **end-to-end trainable selection network**. Motivated by the issue that finding the hyperparameters for the classical approach is cumbersome and that there are more effects at play than the two we considered, we designed a learnable filtering stage. Therefore, we use two soft-attention masks that are learned using a small encoder-decoder-style network. The network gets the image, the depth prior, and the initial stage depth as an input and predicts two independent output masks. The masks have no additional supervision and are only trained through the final loss of the network.

Refinement stage: The final stage takes image and two masked depth maps as an input. The refinement network is a mid-level fusion network, with an image and depth decoder (see Fig. 3). The image encoder processes the input image, and the depth encoder the masked and concatenated depth maps. We perform a deep fusion of the two encoder output feature maps, inspired by [46]. Finally, the decoder processes the fused feature maps and predicts the final depth $\bar{D}$.

D. Dealing with weak supervision

As discussed in Sec. III-B, we train our network in a supervised fashion using a masked regression loss. Give the binary depth mask $M(u, v)$, representing non-zero depth values in the GT we compute our loss as follows,

$$\mathcal{L}_{M,p} = \|M \circ D_{gt} - M \circ \bar{D}\|_p, \tag{4}$$

where $\|\cdot\|_p$ indicates an element-wise p-norm. We use the 1-norm ($p = 1$), other common norms would be the 2-norm or the Huber loss.

Training only with this loss in the case of relative sparse LiDAR ground truth can cause significant artifacts due to the non-uniform supervision by the layered GT. Specifically, the network can learn to only predict the depth at the horizontal scanning lines of the LiDAR. This results in the dense output having a layered-like shape. To avoid this local minimum and generate a reasonable dense depth, we employ two techniques. The first is an edge-aware smoothness loss [47]. The loss penalizes gradients in the depth map but scales the loss according to the image gradient since depth discontinuities often arise near image gradients.

$$\mathcal{L}_{smooth} = \left|\partial_x \hat{D}^*\right| e^{-|\partial_x I|} + \left|\partial_y \hat{D}^*\right| e^{-|\partial_y I|} \tag{5}$$

where $\hat{D}^*$, represents the mean normalized inverse depth, which is used to avoid shrinkage of the depth values.

However, this smoothness loss cannot fully compensate for the strong gradients from the supervised loss. Thus, we add a GAN-based loss, where the discriminator forces the network to produce realistic depth maps. The idea is that the discriminator recognizes LiDAR-like layered depth outputs as fake examples, forcing the generator to avoid this local minimum. In this paper, we use the Least Square GAN (LSGAN) [48] formulation with a PatchGAN [49] discriminator. We consider LSGAN since it forces samples to be closer to the real data, unlike standard GAN-based loss where samples suffer from vanishing gradients. The PatchGAN [49] discriminator is used to give more local feedback to the generator. The GAN-based training is formulated as follows, the generator G produces a depth map $\bar{D}$ given an image I and a sparse depth $\hat{D}$, and we condition the discriminator D on the nearest map image I_{map} and the densified depth map $D_{map,\mathrm{d}}$. Note that we densify the sparse depth map D_{map} using a depth completion network, which is trained using data from the map itself to avoid layering issues in the true examples. Thus, the discriminator has to learn to differentiate (I, $\bar{D}$) and (I_{map}, $D_{map,\mathrm{d}}$) tuples. Thus, the LSGAN objective can be formulated as,

$$\begin{aligned}\mathcal{L}^{\mathrm{D}}_{GAN} = & \frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim p_{\mathrm{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x}|\boldsymbol{y}) - b)^2\right] \\ & + \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}|\boldsymbol{y})) - a)^2\right] \end{aligned} \tag{6}$$

$$\mathcal{L}^{\mathrm{G}}_{GAN} = \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}|\boldsymbol{y})) - c)^2\right] \tag{7}$$

where x represents the pair (I_{map}, $D_{map,\mathrm{d}}$), and z is the predicted depth. Additionally, a are the labels of the fake sample, b the labels of the real samples, and c the value that the generator wants the discriminator to believe for fake data. We set $a = 0$, $b = 1$, and $c = b$ to emphasize the realism of the predicted depth.

Finally, the objective for the generator, our depth fusion network, is given by

$$\mathcal{L}_{total} = \mathcal{L}_{M,1} + \lambda_s \mathcal{L}_{smooth} + \lambda_{GAN} \mathcal{L}^{\mathrm{G}}_{GAN}, \tag{8}$$

which is a combination of the sparse GT loss, edge-aware smoothness loss, and the GAN loss, which are traded of using the hyperparameters $\lambda_s = 0.001$ and $\lambda_{GAN} = 0.01$.

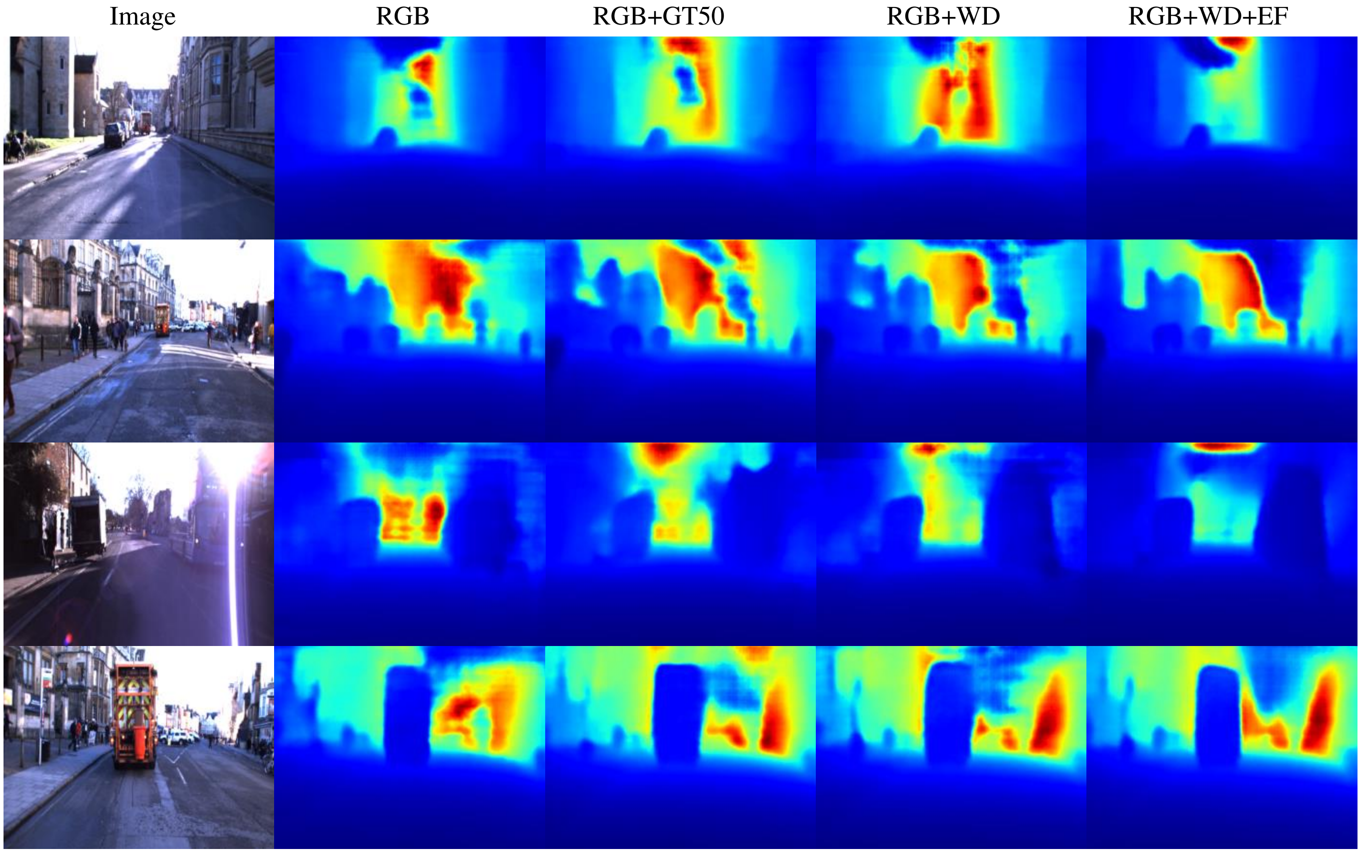

Fig. 6. **The qualitative comparison of our method with baseline methods is shown in this figure.** We can observe that RGB+WD+EF produces sharper and better quality depth maps compared to the other baselines. The method also works in challenging conditions like uneven lighting, lens flares. We can also observe that there are certain artifacts generated in RGB and RGB+WD. This is due to the layered depth pattern in the GT LiDAR depth and also the input in the latter case. Employing a filtering module along with the GAN training helps to alleviate this problem.

IV. Experiments

A. Dataset preparation

The experiments are conducted using the Oxford Radar RobotCar dataset [50], [51]. The dataset contains 32 traversals (routs) of Oxford, UK, equivalent to 280 km of driving. Each route covers (almost) the same streets/area in Oxford, thus the dataset is well suited for our map-based method. Out of these routes, we use one complete route as a “map run”. Further, we select ten routes for training and three routes for testing. The data capture vehicle is mounted with six cameras, two 3D LiDARs, two 2D LiDARs, and a GNSS/INS receiver. We utilize the data from the front-facing camera and 3D LiDARs. We further downsample routes by selecting every 5th frame for training and every 50th frame for testing. The LiDAR data is provided in the form of non-motion-compensated time series data with associated timestamps. We extract the point cloud limited to the camera’s field of view and apply motion compensation to it. We use standard depth estimation metrics for evaluation [26], [29].

B. Implementation details

All the encoders in the initial estimation and refinement stage are a ResNet-18 [52], with the first convolutional layer for the depth encoder changed to deal with the two-channel input. The decoders are similar to [26]. In the classical filtering approach we use α= 5 and β= 18. For the end-to-end learned soft attention masks, we use a lightweight SegNet [53] architecture.

In the map query stage, we use SuperGlue [44] to estimated 2D keypoint matches. We further filter the correspondences using Random Sample Consensus (RANSAC) [54], in the PnP when estimating the relative transformation. Finally, we dilate the sparse depth map using a 5×5 kernel to get the depth of all key points in the PnP. If not stated otherwise, we use only the closest image $N = 1$ to generate the prior map depth $\hat{D}$.

C. Comparison with baselines

In this section, we discuss the qualitative and quantitative results of our method and compare it to baselines and SOTA methods. We show the effectiveness of our approach by achieving the best results amongst compared methods.

We consider a single-stage RGB only method as a minimum baseline (only the initial estimation state in our network) and build all subsequent experiments on top of it as shown in Tab. I. We improve this baseline by including the raw depth D_{map} from the closest map frame (Baseline RGB+D). The network is a simple one-stage depth completion network, which concatenates the depth in the input. To show that aligning the image is important, we train the same model but with the warped prior depth $\hat{D}$ as an input (Baseline RGB+WD). All our baseline networks use the full loss, including the GAN

TABLE I
SUMMARY OF SINGLE DEPTH ESTIMATION/COMPLETION. D: RETRIEVED DEPTH BASED ON LOCATION. WD: WARPED PRIOR DEPTH, CF: CLASSICAL FILTERING, EF: END-TO-END FILTERING

Method	↓ RMSE	↓ MAE	↓ ARD	↑ δ_1	↑ δ_2	↑ δ_3
Baseline methods (B)						
RGB(B)	4.6339	2.1294	0.2185	0.8596	0.9169	0.9397
RGB(B)+D	7.0891	4.2832	0.5874	0.5179	0.7762	0.8586
RGB(B)+WD	4.5171	2.0846	0.2122	0.8621	0.9178	0.9401
SOTA Depth Completion methods						
UncertNet [28]	5.9882	3.2497	0.2843	0.7193	0.8687	0.9167
Sparse2dense [26]	5.2315	2.5814	0.2528	0.7977	0.8949	0.9272
Self Sup.DC [29]	5.0209	2.4677	0.2365	0.8100	0.9030	0.9324
PENet [30]	4.6596	2.2223	0.2236	0.8356	0.9123	0.9381
CSPN [27]	4.6025	2.1403	0.2152	0.8424	0.9150	0.9381
Our method (RGB+WD) with different filtering						
Our (CF)	4.0549	1.7045	0.1893	0.8827	0.9280	0.9425
Our (EF)	**3.8381**	**1.4518**	**0.1749**	**0.8983**	**0.9318**	**0.9440**

loss as discussed in Section III-D. For comparison, we also train several existing SOTA depth completion methods, which receive the warped map depth as an input. Finally, we show the results of our 3-stage network, with both the classical and end-to-end filtering.

We observe that using the not-aligned map depth directly without warping leads to a degradation of the predictions compared to the RGB baseline. However, using the aligned depth with the same network boosts the performance by -0.12m RMSE compared with the RGB baseline. In fact, our baseline network with the warped prior depth as input performs better than all the SOTA depth completion networks with the same inputs. We assume that the better performance is due to the GAN-loss, which helps in the sparse supervision setting. Finally, our 3-stage networks achieve a significant boost in performance compared to the other networks. Already our classical filtering stage results in a 0.46m improvement in RMSE, and learning the filtering gives another 0.22m improvement. The qualitative results are shown in Fig. 6.

D. Ablation study

In this section, we perform ablations on our pipeline. We start with an RGB-only approach and add warped prior depth and our 3-stage network while testing each case with and without the proposed loss functions. In Tab. II, we observe that the GAN-based loss helps to improve the performance in all mentioned cases. Furthermore, it can be observed that the addition of the noisy warped depth in a single-stage network can improve the performance. Finally, adding the end-to-end filtering and refinement stage further boosts performance significantly. We also investigate and conclude that if more than one prior map depth is used to compute $\hat{D}$, the performance benefits are marginal.

We also evaluate how sparse GT depth points in the input of a single-stage network help. The GT depth points are randomly sampled following [26]. We can see that our map-based depth is roughly worth 50 GT points and that, as expected, more GT points can further improve the performance.

TABLE II
SUMMARY OF SINGLE DEPTH ESTIMATION/COMPLETION. GT(N): N GROUND TRUTH SPARSE POINTS, WD: WARPED PRIOR DEPTH, EF: 3-STAGE NETWORK WITH E2E FILTERING, +(N)F: USE OF N ADDITIONAL MAP FRAMES. METHODS BUILD ON TOP OF EACH OTHER.

Input	L_1+L_S	L_{GAN}	↓ RMSE	↓ ARD	↑ δ_1
RGB	✓		4.7169	0.2247	0.8533
	✓	✓	4.6339	0.2185	0.8596
+WD	✓		4.6405	0.2192	0.8597
	✓	✓	4.5171	0.2122	0.8621
+EF	✓		3.8506	0.1891	0.8865
	✓	✓	**3.8381**	**0.1749**	**0.8983**
+3F	✓	✓	3.7936	0.1720	0.8995
+5F	✓	✓	**3.7702**	**0.1693**	**0.9050**
RGB+GT50	✓		3.8426	0.1757	0.8930
RGB+GT1000	✓		**2.9462**	**0.1599**	**0.9185**

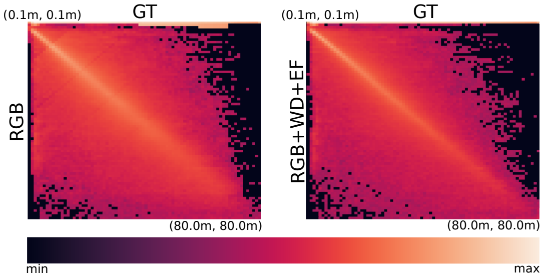

Fig. 7. **Confusion matrix comparing RGB-only vs GT depth and RGB+WD+EF vs GT depth.** The spread of the diagonal shows that RGB+WD+EF performs significantly better than RGB-only for larger depth ranges.

Scale of predicted depth. One of the reasons for using prior depth data is to generate the depth maps to the absolute metric scale. Here we can evaluate the predicted depth using a confusion matrix (Fig 7). The spread of the diagonal of the matrix provides an intuition of uncertainty in the metric scale prediction. We can observe that in the case of RGB-only based predictions, the spread of the diagonal funnels out and disappears as we move towards larger depth values, whereas our proposed method performs significantly better for larger depth ranges.

Effectiveness of method in Depth Completion. We also test our approach in a depth completion setup for completeness and fairness. Here, the input depth is generated by randomly sampling 1000 GT depth points, i.e. a density of less than 33%. We observe that SOTA depth completion methods outperforms our method. The method [30] provides the best results in this setup followed by [28] and [29]. A drop of ∼0.6m in the RMSE metric, ∼0.05m in terms of ARD and ∼3% in δ_1 is expected as our method is designed to be robust against noisy depth inputs as explained in Sec. IV-C. Here, our filtering module becomes largely ineffective leaving the performance solely to the depth prediction networks. We believe that adapting our pipeline to improved depth prediction networks will substantially improve the performance in this setting.

V. CONCLUSION

In this work, we introduce a novel method to enhance the monocular depth estimation method with the help of a map-based prior depth. We propose a simple map generation method that can be added to existing datasets and allows us to show the advantages of using map-based depth information. We observe that just including warped prior depth in the input along with the image provides a considerable performance improvement over an image-only method. These improvements are further confirmed by our proposed 3-stage depth fusion network. The pipeline robustly handles dynamic objects in the scene as well as misalignment in localization, generating SOTA results. Our method is suitable as a starting point for monocular camera-based depth prediction algorithms to further improve the results by utilizing HD-map information.

References

[1] A. Pare, S. Zhang, and Z. Lei, "Multipath interference suppression in time-of-flight sensors by exploiting the amplitude envelope of the transmission signal," *IEEE Access*, 2020. 1

[2] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map pred. from a single image using a multi-scale deep net." in *NIPS*, 2014. 1

[3] C. Sakaridis, D. Dai, and L. Van Gool, "Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," *T-PAMI*, 2020. 1

[4] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting hd maps for 3d object detection," in *CoRL*, 2018. 1

[5] K. Irie and M. Tomono, "Road recognition from a single image using prior information," in *IROS*, 2013. 1

[6] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019. 2

[7] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *T-PAMI*, 2018. 2

[8] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *T-PAMI*, 2009. 2

[9] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *ECCV*, 2016. 2

[10] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017. 2

[11] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," 2015. 2

[12] A. B. Vasudevan, D. Dai, , and L. Van Gool, "Semantic object prediction and spatial sound super-resolution with binaural sounds," *ECCV*, 2020. 2

[13] A. CS Kumar, S. M. Bhandarkar, and M. Prasad, "Monocular depth prediction using generative adversarial networks," in *CVPRW*, 2018. 2

[14] M. Zhu, M. Ghaffari, Y. Zhong, P. Lu, Z. Cao, R. M. Eustice, and H. Peng, "Monocular depth prediction through continuous 3d loss," in *IROS*, 2020. 2

[15] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014. 2

[16] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015. 2

[17] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016. 2

[18] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *T-PAMI*, 2016. 2

[19] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *CVPR*, 2017. 2

[20] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017. 2

[21] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018. 2

[22] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018. 2

[23] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *CVPR*, 2018. 2

[24] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *ECCV*, 2018. 2

[25] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *3DV*, 2017. 2

[26] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *ICRA*, 2018. 2, 6, 7

[27] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *ECCV*, 2018. 2, 7

[28] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *MVA*, 2019. 2, 7

[29] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *ICRA*, 2019. 2, 6, 7

[30] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Towards precise and efficient image guided depth completion," in *ICRA*, 2021. 2, 7

[31] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *CVPR*, 2019. 2

[32] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," in *CVPR*, 2020. 2

[33] V. Patil, W. Van Gansbeke, D. Dai, and L. Van Gool, "Don't forget the past: Recurrent depth estimation from monocular video," *RA-L*, 2020. 2

[34] J.-T. Lin, D. Dai, and L. Van Gool, "Depth estimation from monocular images and sparse radar data," in *IROS*, 2020. 2

[35] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, "Sparse sensing for resource-constrained depth reconstruction," in *IROS*, 2016. 2

[36] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *ICRA*, 2017. 2

[37] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *ICRA*, 2018. 2

[38] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from rgb and sparse sensing," in *ECCV*, 2018. 2

[39] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *T-PAMI*, 2014. 2

[40] Y. Li, L. Sun, and T. Xue, "Fast frame-rate up-conversion of depth video via video coding," in *ACM-MM*, 2011. 2

[41] J. Noraky and V. Sze, "Depth map estimation of dynamic scenes using prior depth information," *arXiv*, 2020. 2

[42] H.-M. Wang, C.-H. Huang, and J.-F. Yang, "Depth maps interpolation from existing pairs of keyframes and depth maps for 3d video generation," in *ISCAS*, 2010. 2

[43] K. Vanhoey, C. E. P. de Oliveira, H. Riemenschneider, A. Bódis-Szomorú, S. Manén, D. P. Paudel, M. Gygli, N. Kobyshev, T. Kroeger, D. Dai, *et al.*, "Varcity-the video: the struggles and triumphs of leveraging fundamental research results in a graphics video production," in *ACM SIGGRAPH Talks*, 2017. 3

[44] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020. 3, 6

[45] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018. 4

[46] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (ddp) from single image and sparse range," in *CVPR*, 2019. 5

[47] P. Heise, S. Klose, B. Jensen, and A. Knoll, "Pm-huber: Patchmatch with huber regularization for stereo matching," in *ICCV*, 2013. 5

[48] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *ICCV*, 2017. 5

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017. 5

[50] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *ICRA*, 2020. 6

[51] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *IJRR*, 2017. 6

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 6

[53] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *T-PAMI*, 2017. 6

[54] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, 1981. 6