2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

UnRectDepthNet: Self-Supervised Monocular Depth Estimation using a Generic Framework for Handling Common Camera Distortion Models

Varun Ravi Kumar[1,4], Senthil Yogamani[2], Markus Bach[1],
Christian Witt[1], Stefan Milz[3,4] and Patrick Mäder[4]
[1]Valeo DAR Kronach, Germany [2]Valeo Vision Systems, Ireland
[3]Spleenlab GmbH, Germany [4]Technische Universität Ilmenau, Germany

***Abstract*— In classical computer vision, rectification is an integral part of multi-view depth estimation. It typically includes epipolar rectification and lens distortion correction. This process simplifies the depth estimation significantly, and thus it has been adopted in CNN approaches. However, rectification has several side effects, including a reduced field of view (FOV), resampling distortion, and sensitivity to calibration errors. The effects are particularly pronounced in case of significant distortion (e.g., wide-angle fisheye cameras). In this paper, we propose a generic scale-aware self-supervised pipeline for estimating depth, euclidean distance, and visual odometry from unrectified monocular videos. We demonstrate a similar level of precision on the unrectified KITTI dataset with barrel distortion comparable to the rectified KITTI dataset. The intuition being that the rectification step can be implicitly absorbed within the CNN model, which learns the distortion model without increasing complexity. Our approach does not suffer from a reduced field of view and avoids computational costs for rectification at inference time. To further illustrate the general applicability of the proposed framework, we apply it to wide-angle fisheye cameras with 190° horizontal field of view. The training framework *UnRectDepthNet* takes in the camera distortion model as an argument and adapts projection and unprojection functions accordingly. The proposed algorithm is evaluated further on the KITTI rectified dataset, and we achieve state-of-the-art results that improve upon our previous work FisheyeDistanceNet [1]. Qualitative results on a distorted test scene video sequence indicate excellent performance[1].**

I. Introduction

Depth estimation is a crucial task for automated driving, and multi-view geometric approaches were traditionally used for computing depth. Some of the initial prototypes of automated driving relied primarily on depth estimation [2], and to enable accurate depth estimation, stereo cameras were used. CNN models have been dominant with supervised learning in various visual perception tasks. Self-supervised learning has enabled high accuracy in depth estimation [3], [4], [5], [6], [7]. There is also a trend of integrating depth estimation task into multi-task models [8], [9], [10]. Most of the depth estimation methods were demonstrated in the context of automated driving on rectified KITTI video sequences where barrel distortion was removed.

Rectification is considered to be a fundamental step in dense depth estimation [11]. In the case of stereo cameras, epipolar rectification is performed to enable matching only in one direction along the horizontal scanline. This approach can also be extended to monocular cameras by using two consecutive frames giving rise to motion stereo. These rectification steps also require the removal of non-linear distortion. Although it is convenient to work with rectilinear projections, there are practical issues that arise due to rectification, which are discussed in detail in Section II. Rectification has also been transferred to CNN based approaches as an inductive bias to simplify the learning. Yadati *et al.* [12] demonstrate that CNN based two-view depth estimation is challenging without rectification and attempt to solve it in a simpler setting. To the best of our knowledge, all the methods reported on KITTI make use of the barrel distortion corrected images. Automotive cameras like fisheye surround-view cameras exhibit a strong distortion, and it is not easy to rectify their images. Recently, several tasks such as motion segmentation [13], and soiling detection [14] were demonstrated on fisheye images without rectification.

Fig. 1 **Depth obtained from a single unrectified (left) and rectified KITTI image (right).** Our scale-aware model, UnRectDepthNet, yields precise boundaries and fine-grained depth maps.

In our previous article, FisheyeDistanceNet [1], we introduced the first end-to-end scale-aware, self-supervised monocular training method for fisheye cameras with a large field of view to regress a Euclidean distance map. In this work, we generalize the training framework to work with any camera model and propose a fully-differentiable architecture that estimates the depth directly from raw unrectified images (shown in Fig. 3) without the need for any pre-processing. In terms of the motivation of a generic training pipeline, the closest related work is CAM-Convs [15], where authors propose a generic framework for different types of cameras. However, they do not handle non-linear distortion, and it is not a self-supervised method. Our contributions include:

- A novel generic end-to-end self-supervised training

[1]https://youtu.be/K6pbx3bU4Ss

pipeline to estimate monocular depth maps on raw distorted images for various camera models.
- Empirical evaluation of our approach on two diverse automotive datasets, namely KITTI and WoodScape.
- First demonstration of depth estimation results directly on unrectified KITTI sequences (see Fig. 1).
- State-of-the-art results on KITTI depth estimation among self-supervised methods.

II. Motivation for Working on Raw Images

Distortion in Automotive Cameras: To handle the wide variety of automotive use cases, different cameras having a different field of view are used. The most common ones are around $100°$ hFOV (horizontal field of view) cameras used for front camera sensing and $190°$ hFOV fisheye lens cameras for surround-view sensing. Due to their moderate to large FOV, these cameras suffer from lens distortion, whose main component is typically radial distortion and minor tangential distortion.

Moderate FOV Lens Model: For lenses with a moderate FOV ($< 120°$), Brown–Conrady model [16] is commonly used as it models both radial and tangential distortion. For larger FOV, this distortion model typically breaks down or requires very high polynomial orders. The KITTI dataset's calibration uses this model based on OpenCV's implementation. In this model, the projection function $X_c \mapsto \Pi(X_c) = p$ maps a 3D point $X_c = (x_c, y_c, z_c)^T$ in camera coordinates to a pixel $p = (i, j)^T$ in the image coordinates. It is calculated in the following way:

$$x = x_c/z_c, \quad y = y_c/z_c$$
$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$
$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy$$
$$i = f_x \cdot x' + c_x, \quad j = f_y \cdot y' + c_y$$

where k_1, k_2, and k_3 are radial distortion coefficients, p_1 and p_2 are tangential distortion coefficients of the lens, $r^2 = x^2 + y^2$, f_x, f_y are the focal lengths and c_x, c_y are the coordinates of the principal point.

Fisheye Lens Models: For fisheye lenses, the mapping of 3D points to pixels universally requires a radial component $r(\theta)$ [17]. The projection is a complex multi-stage process compared to regular lenses and thus we list the detailed steps:

1) The point X_c in camera coordinates is mapped to a unit vector as $S = (s_x, s_y, s_z)^T = X_c/\|X_c\|$.
2) The incident angle against the optical axis (coincident with the Z-axis) $\theta = \frac{\pi}{2} - \arcsin(s_z)$ is computed.
3) The radial function $r(\theta)$ to get the radius on the image plane (typically in pixels) is computed.
4) Given the pixel distortion centre (c_x, c_y), the pixel location is given by $i = r \cdot s_x/\rho + c_x$ and $j = r \cdot s_y/\rho + c_y$ with $\rho = \sqrt{s_x^2 + s_y^2}$.
5) (optional) Depending on the model used in Step 3, an additional distortion correction may need to be applied.

We discuss the projection models which are supported in our framework. The polynomial model is the commonly used one and relatively recent projection models are UCM (Unified Camera Model) [18] and eUCM (Enhanced UCM) [19]. Rectilinear (representation of pinhole model) and Stereographic (mapping of sphere to a plane) are not suitable for fisheye lenses but provided for comparison. Double Sphere [20] is a recently proposed model which has a closed form inverse with low computational complexity. The radial distortion models are summarized below:

1) Polynomial: $r(\theta) = a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4$
2) UCM: $r(\theta) = f \cdot \sin\theta/(\cos\theta + \xi)$
3) eUCM: $r(\theta) = f \cdot \frac{\sin\theta}{\cos\theta + \alpha\left(\sqrt{\beta \cdot \sin^2\theta + \cos^2\theta} - \cos\theta\right)}$
4) Rectilinear: $r(\theta) = f \cdot \tan\theta$
5) Stereographic: $r(\theta) = 2f \cdot \tan(\theta/2)$
6) Double Sphere: $r(\theta) = f \cdot \frac{\sin\theta}{\alpha\sqrt{\sin^2\theta + (\xi + \cos\theta)^2} + (1-\alpha)(\xi + \cos\theta)}$

Fig. 2 **Illustration of distortion correction in KITTI and WoodScape datasets.** The first row shows a raw KITTI image with barrel distortion and the corresponding rectified image. The red box was used to crop out black pixels in periphery causing a loss of FOV. The second row shows a raw WoodScape image with strong fisheye lens distortion and the corresponding rectified image exhibiting a drastic loss of FOV.

Practical Problems encountered: In the previous subsection, we have established that real-world automotive cameras have lens distortion. The typical approach is to remove the distortion and then apply standard camera projection models. However, in practice, this has several issues that are not dealt with in the literature. Fig. 2 illustrates the rectification used in KITTI and WoodScape datasets. In the KITTI dataset, due to the barrel[2] distortion effect of the camera, images have been rectified and cropped to 1242×375 pixels. Cropping is performed after rectification to get a rectangular grid without any black pixels in the periphery. Thus the size of the rectified images is smaller than that of the raw images with 1392×512 pixels. Based on the number of the non-black, occupied pixels removed by the cropping, roughly 10% of the image information is lost. This effect becomes more drastic for the WoodScape images with a much larger radial distortion were more than 30% of the image information is lost[3]. For a horizontal FOV greater than $180°$, there are rays incident from behind the camera, making it theoretically

[2] KITTI [21] refers to it as pincushion distortion because of an error in the OpenCV documentation which was fixed later.

[3] Other quasi-linear rectification methods like cylindrical rectification will preserve more information at the cost of additional distortion.

impossible to establish a complete mapping to a rectilinear viewport. Thus the rectification defeats the purpose of using a wide-angle fisheye lens.

Reduced FOV is the most critical problem of undistortion, but there are further practical issues. The first one is resampling distortion, which is caused by interpolation errors during the warping step. This effect can be partially mitigated by a more advanced interpolation method [22]. However, it is particularly strong in the periphery of fisheye lenses because a small region is expanded to a larger one in the warped image. Besides, the warping step is needed at inference time, which consumes significant computing power and memory bandwidth.

The other issue is related to calibration. In an industrial setup, millions of cameras are deployed, and they have manufacturing variations. The camera parameters (mainly focal length) can also vary due to high ambient temperatures when driving in a hot region. Thus a model that relies on rectification to correct the distortion could have errors. For instance, dataset capture and training are typically performed on one particular camera, and the model is deployed to work on millions of cameras in commercial vehicles. Thus rectification and cropping to a standard resolution as per the training camera are sub-optimal for a deployed camera. However, if CNN learns the distortion as part of the transfer function, it is only weakly encoded and thus expected to be more robust. To alleviate these issues, we are motivated to explore a depth estimation model that can work directly on raw images without needing rectification.

III. Self-Supervised Scale-Aware Depth Estimation

Following Zhou *et al.* [4] we aim at learning a self-supervised monocular structure-from-motion (SfM):

1) a scale-ambiguous depth $\hat{D}$ is obtained through a self-supervised monocular model $g_D : I_t \rightarrow D$ outputting $\hat{D} = g_D(p)$ per pixel p in the target image I_t; and
2) Rigid transformations $T_{t\rightarrow t'} \in \mathrm{SE}(3)$ which comprise a set of 6 degrees of freedom are estimated using an ego-motion predictor $g_x : (I_t, I_{t'}) \rightarrow I_{t\rightarrow t'}$, between the target image I_t and the set of reference images $I_{t'}$. Specifically, $t' \in \{t+1, t-1\}$, i.e. the frames I_{t-1} and I_{t+1} are used as reference images, but it would in general be possible to use a larger offset from the temporal consistent sequence.

In all the previous works [4], [3], [23], networks are equipped to retrieve inverse depth $g_d : p \mapsto 1/g_D(p)$. One downside to these methods is the scale ambiguity in both depth and pose estimation. In this work, we recover scale-aware depth directly for distorted images. View-synthesis is used as a self-supervising technique, and the network is trained with the source images I_{t-1} and I_{t+1} to synthesize the appearance of a target image I_t. For this, we need the projection function Π of the chosen camera model, which maps a 3D point X_c in camera coordinates to a pixel $p = \Pi(X_c)$ in image coordinates. An overview of projection models for different lens types can be found in Section II. The corresponding unprojection function Π^{-1}, which maps an image pixel p and its depth estimate $\hat{D}$ to the 3D point $X_c = \Pi^{-1}(p, \hat{D})$, is also required. If Π^{-1} cannot be expressed in analytic form, a pre-calculated lookup table is used to ensure computational efficiency.

View Synthesis for Various Camera Models: Using the network's $\hat{D}_t$ depth estimate for frame I_t at time t a point cloud P_t is obtained through:

$$P_t = \Pi^{-1}(p_t, \hat{D}_t) \tag{1}$$

Here, the pixel set of image I_t is represented by p_t and Π^{-1} denotes the unprojection function introduced above. The pose of target image I_t relative to the pose of the source image $I_{t'}$ is considered as $T_{t\rightarrow t'}$ and estimated by the pose network. For the point cloud of frame $I_{t'}$ an estimate $\hat{P}_{t'} = T_{t\rightarrow t'} P_t$ is obtained by applying this transformation. The projection model Π is used at time t' to project $\hat{P}_{t'}$ onto the camera. The projection and transformation are combined with Eq. 1 to establish a mapping between the image coordinates $p_t = (i, j)^T$ at time t and $\hat{p}_{t'} = (\hat{i}, \hat{j})^T$ at time t'. A view-synthesized reconstruction $\hat{I}_{t'\rightarrow t}$ of the target frame I_t is computed via a backward warp of the source frame $I_{t'}$ using this mapping.

$$\hat{p}_{t'} = \Pi\big(T_{t\rightarrow t'} \Pi^{-1}(p_t, \hat{D}_t)\big), \quad \hat{I}^{ij}_{t'\rightarrow t} = \left\langle I^{\hat{i}\hat{j}}_{t'} \right\rangle \tag{2}$$

Because the warped coordinates $\hat{i}, \hat{j}$ are continuous, differentiable spatial transformer network [24] can be used to synthesize $\hat{I}_{t'\rightarrow t}$ by bilinear interpolation of the four pixels of $I_{t'}$ nearby $\hat{p}_{t'}$. The $\langle \dots \rangle$ symbol denotes the corresponding operator used for sampling.

Reconstruction Loss: The L1 pixel-wise loss is coupled with Structural Similarity (SSIM) [25], to form an image reconstruction loss $\mathcal{L}_r$ between the target image I_t and the reconstructed target image $\hat{I}_{t'\rightarrow t}$ given by Eq. 3 below.

$$\begin{aligned} \tilde{\mathcal{L}}_r(I_t, \hat{I}_{t'\rightarrow t}) &= \omega \, \frac{1 - \mathrm{SSIM}(I_t, \hat{I}_{t'\rightarrow t}, \mathcal{M}_{t\rightarrow t'})}{2} \\ &\quad + (1-\omega) \, \Big\| (I_t - \hat{I}_{t'\rightarrow t}) \odot \mathcal{M}_{t\rightarrow t'} \Big\|_{l^1} \\ \mathcal{L}_r &= \min_{t' \in \{t+1, t-1\}} \tilde{\mathcal{L}}_r(I_t, \hat{I}_{t'\rightarrow t}) \end{aligned} \tag{3}$$

The binary mask $\mathcal{M}_{t\rightarrow t'}$ from [1] is incorporated, element-wise multiplication is denoted by $\odot$ and ω is set to 0.85. Following [3], we adopt a per-pixel minimum compared to averaging over all the source images. This yields higher accuracy by reducing the artifacts and significantly sharpening the boundaries of occlusion. We clip the reconstruction loss values to a 95^{th} percentile based on [26] to diminish the impact of dynamic objects or occluded areas in the scene. It indirectly aids the optimization algorithm to have a robust reconstruction error.

Solving Scale Factor Ambiguity at Training Time: The network's sigmoid output σ can be translated to depth with $D = 1/(x\sigma + y)$ based on the fact that $depth \propto 1/disparity$ for a rectified pinhole projection model, where x and y are chosen to constrain D between 0.1 and 100 units [3]. We can only obtain angular disparities [27] for

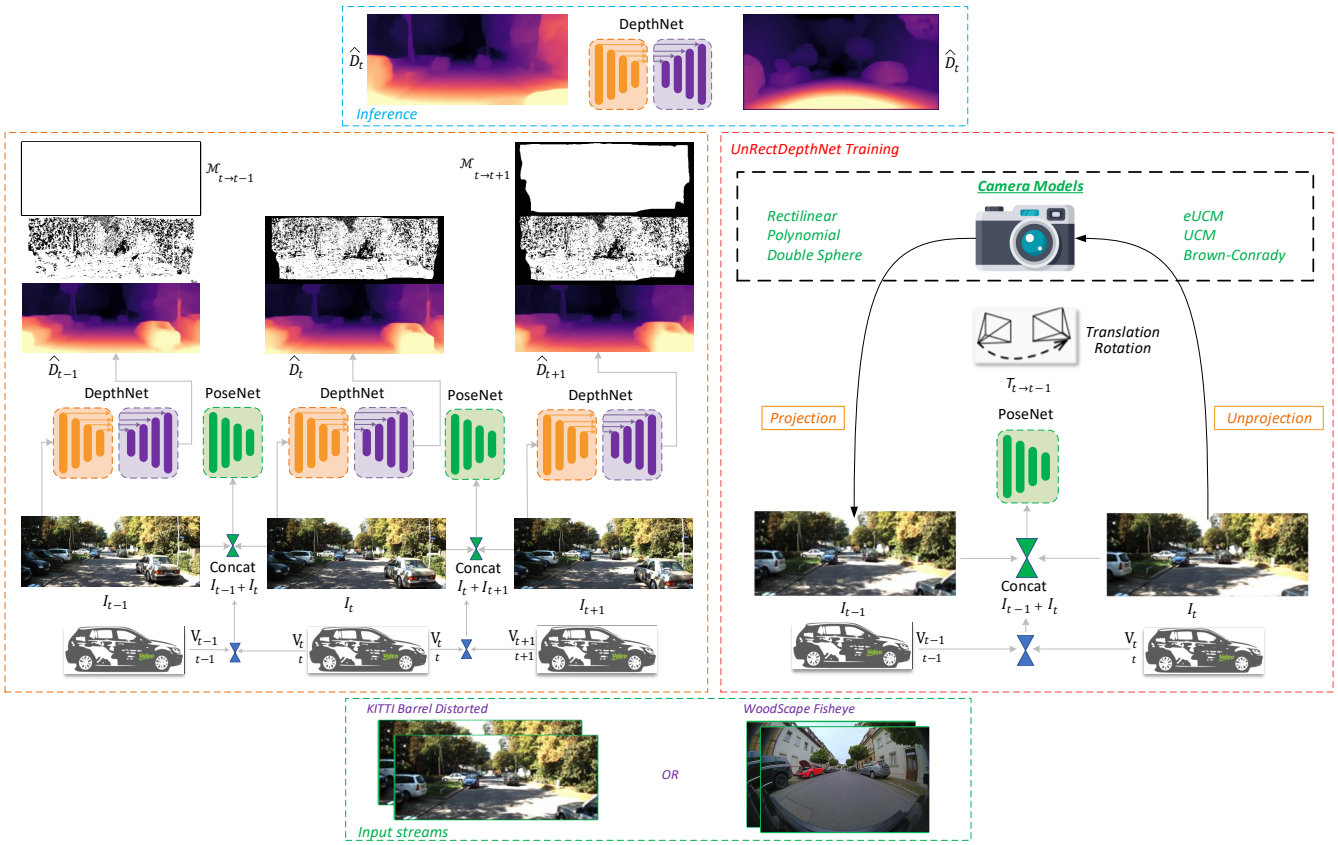

Fig. 3: **Overview of our generic depth estimation training framework UnRectDepthNet.** The UnRectDepthNet training block on the right enables the usage of various camera models generically listed in the black box. The distortion is then handled internally in the unprojection and projection steps of the transformation from I_t to I_{t-1}. In this paper, we have tested it with KITTI barrel distorted and WoodScape fisheye distorted video sequences. The block on the left indicates the entire workflow of the training pipeline where the top row depicts the ego masks as explained in [1], $\mathcal{M}_{t\to t-1}$, $\mathcal{M}_{t\to t+1}$ representing the valid pixel coordinates while synthesizing $\hat{I}_{t-1\to t}$ from I_{t-1} and $\hat{I}_{t+1\to t}$ from I_{t+1} respectively. The following row showcases the masks used to filter static pixels, obtained after training two epochs, and the black pixels are removed from the reconstruction loss. Dynamic objects moving at speed similar to the ego car's as well as homogeneous areas are filtered out to prevent the contamination of reconstruction loss. The third row shows the depth predictions, where the scale ambiguity is resolved using the ego vehicle's odometry data. Finally, the top block illustrates the inference output.

camera models which undergo distortion. To perform a successful inverse warp operation of source images $I_{t'}$ onto the target frame I_t, metric depth values are required. *Scale-ambiguous* estimates from both the g_d monocular depth model and the $g_{\mathbf{x}}$ ego-motion predictor is a hindrance to the obtainment of metric depth maps on any camera model of choice because of the inherent drawback of the self-supervised structure-from-motion objective. Following [1], we solve the scale ambiguity by normalizing the pose network's prediction $T_{t\to t'}$ to obtain scale-aware depth values. We compute the displacement magnitude Δx relative to target frame I_t utilizing the ego vehicle's instantaneous velocity predictions $v_{t'}$ at time t' and v_t at time t obtained from its odometry data. Finally, we scale the normalized translation vector with Δx. The same method is also incorporated on KITTI [28] rectified pinhole dataset to achieve scale-aware depth maps.

$$\overline{T}_{t\to t'} = \frac{T_{t\to t'}}{\|T_{t\to t'}\|} \cdot \Delta x \tag{4}$$

Edge-Aware Depth Smoothness Loss: A geometric smoothness loss is added to regularize depth and avoid different values in occluded or homogeneous areas. We incorporate the edge-aware loss term and impose it on the inverse depth map similar to [5], [29], [30].

$$\mathcal{L}_s(\hat{D}_t) = |\partial_i \hat{D}_t^*| e^{-|\partial_i I_t|} + |\partial_j \hat{D}_t^*| e^{-|\partial_j I_t|} \tag{5}$$

Following [6], mean-normalized inverse depth $\hat{D}_t^*$ of the target image I_t is considered to avoid any shrinkage of depth estimates $\hat{D}_t$, i.e. $\hat{D}_t^* = \hat{D}_t^{-1}/\overline{D}_t$, where $\overline{D}_t$ denotes the mean of $\hat{D}_t^{-1} := 1/\hat{D}_t$. ***Final Training Loss:*** The final self-supervised structure-from-motion (SfM) objective comprises a reconstruction loss $\mathcal{L}_r$ applied on forward and backward sequences and an edge-aware smoothness term $\mathcal{L}_s$ to regularize depth. Additionally, $\mathcal{L}_{dc}$ a cross-sequence depth consistency loss estimated from the sequence of frames in the training videos is also incorporated just as in [1]. Since the bilinear sampler has gradient locality [24], we include four scales for training as suggested in [4], [5] mainly to reduce the chance of training reaching a local minimum. The overall objective function is averaged over the number

	Method	Resolution	Dataset	Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
				lower is better				higher is better		
Original [31]	SfMLeaner [4]	416 x 128	K	0.183	1.595	6.709	0.270	0.734	0.902	0.959
	Vid2depth [29]	416 x 128	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	DDVO [6]	416 x 128	K	0.151	1.257	5.583	0.228	0.810	0.936	0.974
	EPC++ [32]	640 x 192	K	0.141	1.029	5.350	0.216	0.816	0.941	0.976
	Struct2Depth [33]	416 x 128	K	0.141	1.026	5.291	0.215	0.816	0.945	0.979
	Monodepth2 [3]	640 x 192	K	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	PackNet-SfM [23]	640 x 192	K	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	Monodepth2 [3]	1024 x 320	K	0.115	0.882	4.701	0.190	0.879	0.961	0.982
	UnRectDepthNet	640 x 192	K	**0.107**	**0.721**	**4.564**	**0.178**	**0.894**	**0.971**	**0.986**
	UnRectDepthNet	1024 x 320	K	0.103	0.705	4.386	0.164	0.897	0.980	0.989
	UnRectDepthNet	608 x 224	KD	0.102	0.720	4.559	0.183	0.892	0.973	0.988
	UnRectDepthNet	1216 x 448	KD	0.106	0.709	4.357	0.161	0.895	0.984	0.992
	FisheyeDistanceNet [1]	512 x 256	WS	0.152	0.768	2.723	**0.210**	0.812	0.954	0.974
	UnRectDepthNet	512 x 256	WS	**0.148**	**0.702**	**2.530**	0.212	**0.826**	**0.960**	**0.980**
Improved [34]	SfMLeaner [4]	416 x 128	K	0.176	1.532	6.129	0.244	0.758	0.921	0.971
	Vid2Depth [29]	416 x 128	K	0.134	0.983	5.501	0.203	0.827	0.944	0.981
	DDVO [6]	416 x 128	K	0.126	0.866	4.932	0.185	0.851	0.958	0.986
	EPC++ [32]	640 x 192	K	0.120	0.789	4.755	0.177	0.856	0.961	0.987
	Monodepth2 [3]	640 x 192	K	0.090	0.545	3.942	0.137	0.914	0.983	0.995
	PackNet-SfM [23]	640 x 192	K	**0.078**	0.420	3.485	0.121	**0.931**	0.986	**0.996**
	UnRectDepthNet	640 x 192	K	0.081	**0.414**	**3.412**	**0.117**	0.926	**0.987**	**0.996**
	UnRectDepthNet	640 x 224	KD	0.092	0.458	3.503	0.132	0.906	0.971	0.990

TABLE I: **Quantitative performance comparison of UnRectDepthNet** for depths up to 80 m for KITTI and 40 m for WoodScape. In the Dataset column, K refers to KITTI [28], KD refers to the KITTI distorted [21], and WS refers to WoodScape [35] dataset. *Original* refers to depth maps defined in [31], and *Improved* refers to refined depth maps provided by [34]. All the methods listed in the table are self-supervised approaches on monocular camera sequences. At inference time, all the approaches except UnRectDepthNet and PackNet-SfM scale the estimated depths using median ground-truth LiDAR depth. We generalized our previous model FisheyeDistanceNet in our new training framework and added additional features which improve results on WoodScape.

Method	FS	BS	SR	CSDCL	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours	✓	✓	✓	✓	0.102	0.720	4.559	0.183	0.892	0.973	0.988
Ours	✓		✓	✓	0.131	0.856	4.933	0.198	0.853	0.954	0.968
Ours	✓			✓	0.141	0.971	5.183	0.206	0.831	0.941	0.953
Ours	✓				0.144	1.011	5.204	0.225	0.822	0.945	0.949

TABLE II: **Ablation study of our algorithm** on the KITTI Eigen split dataset [36]. Depths are capped at 80 m. FS, BS, SR, CSDCL indicate forward sequence, backward sequence, super-resolution network with PixelShuffle [37] layers and cross-sequence depth consistency loss, respectively. The input resolution is 608×224 pixels.

of pixels, scales and batches.

$$\mathcal{L} = \sum_{n=1}^{4} \frac{\mathcal{L}_n}{2^{n-1}}, \tag{6}$$

$$\mathcal{L}_n = {}^n\mathcal{L}_r^f + {}^n\mathcal{L}_r^b + \gamma\ {}^n\mathcal{L}_{dc} + \beta\ {}^n\mathcal{L}_s$$

IV. Implementation Details

The depth estimation network is mainly based on FisheyeDistanceNet [1]. We use Pytorch [38] and employ Ranger (RAdam [39] + LookAhead [40]) optimizer to minimize the training objective function (6). The model is trained using Titan RTX with a batch size of 20 for 20 epochs, with initial learning rate of 4 x 10^{-4} with OneCycleScheduler [41]. The network's sigmoid output σ is converted to depth with $D = 1/(x \cdot \sigma + y)$ for pinhole model and $D = x \cdot \sigma + y$ for fisheye, where x and y are chosen such that D is bounded between 0.1 and 100 units. For KITTI distorted images, we use 608×224 pixels, and for WoodScape fisheye images 512×256 pixels as the network input to maintain the original aspect ratio. The loss weighting factors β and γ of smoothness and cross-sequence depth consistency loss are set to 0.001. To remove checkerboard artifacts in the sub-pixel convolution [42], the final convolutional layers are initialized in a particular manner before the pixel shuffle operation as described in [37].

V. Experiments

Datasets – KITTI and WoodScape: For experiments with the Brown–Conrady model, the KITTI dataset is used as per the split defined by Eigen *et al.* [36]. Following Zhou *et al.* [4], the static frames are dropped from the dataset. There are 39,810 images for training, 4,424 images for validation, and 697 images for testing. We also make use of the 652 test frames from the Eigen split with improved ground truth provided by [34]. The WoodScape [35] dataset distribution can be found in our FisheyeDistanceNet paper [1].

Evaluation: To facilitate the comparison, we evaluate the results of UnRectDepthNet's depth estimation using the metrics proposed by Eigen *et al.* [31]. Table I and Fig. 4 indicate the quantitative and qualitative results. The former illustrates that our scale-aware self-supervised approach on KITTI rectified outperforms almost all the state-of-the-art monocular approaches and the KITTI distorted results are better than most of the previous outcomes obtained with self-supervised approaches on the corresponding rectified dataset.

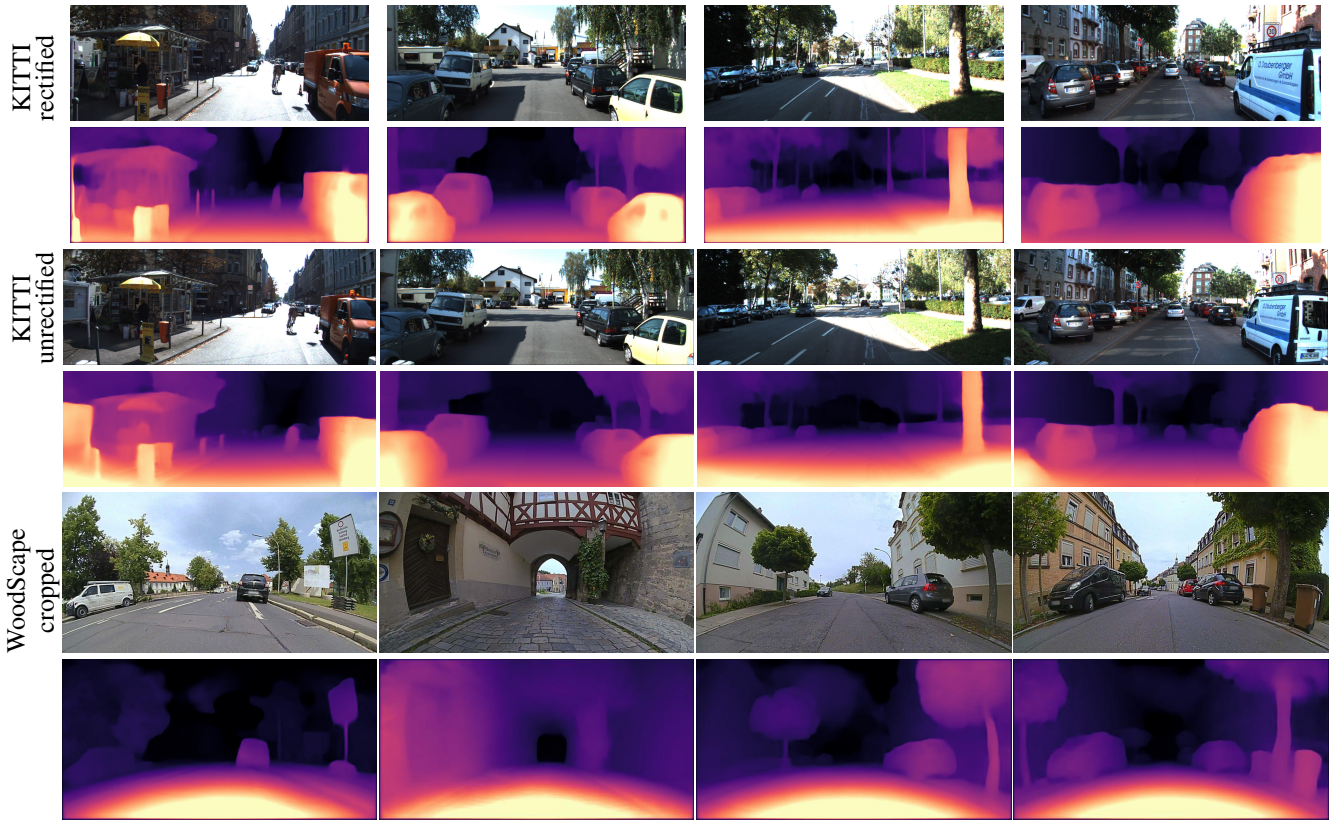

Fig. 4: **Qualitative result comparison on KITTI and WoodScape dataset.** The results on a distorted test video sequence indicate excellent performance, see https://youtu.be/K6pbx3bU4Ss.

Owing to the absence of odometry data, the Cityscapes dataset is not leveraged into our training framework.

Because the projection's operators are different, prior approaches to depth estimation will not work on the WoodScape fisheye dataset without significant redesign to incorporate fisheye projection geometry. In addition, fisheye cameras are designed for near-field sensing, and we only compare up to a range of 40 m as per FisheyeDistanceNet [1]. We generalized the training methodology of this model to incorporate any arbitrary distortion model. We also tuned our network to the optimal hyperparameters using grid search and removed batch normalization in the decoder as we observed ghosting effects and holes in homogeneous areas. We calculated the minimum reconstruction error for the two warps of the backward sequence individually compared to a combined minimization for forward sequence since here the target frames are $I_{t'}$ ($t' \in \{t+1, t-1\}$), as explained in [1].

KITTI Distorted Ablation Study: We perform an ablation study to understand the significance of different components used and tabulate in Table II: (i) *Remove Backward Sequence*: The network is trained only for a forward sequence consisting of two warps, as explained in [1]. The impact is significant in the border areas as fewer constraints are induced. The model inherently fails to resolve unknown depths in those areas at the test time, which was also observed in previous works [3], [4], [43]; (ii) *Additionally remove Super-Resolution using sub-pixel convolution*: It has a significant effect as distant objects are small in fisheye cameras and cannot be resolved correctly with simple nearest-neighbor interpolation or transposed convolution [44]; (iii) *Additionally remove cross-sequence depth consistency loss*: The removal of the CSDCL diminishes the baseline, induces fewer constraints, and the model is not robust to yield accurate depth estimates.

VI. Conclusion

We introduced a generic self-supervised training method for depth estimation handling distorted images. We support various commonly used automotive camera models in the framework and indicate empirical results on KITTI and WoodScape datasets. For KITTI, we show that depth estimation on unrectified images can produce the same accuracy as on rectified images. We also obtain a state-of-the-art result on KITTI among self-supervised algorithms. The same framework was used to train with fisheye unrectified images from the WoodScape dataset, where only the corresponding camera model parameters were updated. In future work, we aim to extend the training framework to take in various camera streams as input and output a generic inference model which can take in the camera model as argument.

Acknowledgements: We want to thank Valeo, especially DAR Kronach, Germany and Valeo Vision Systems, Ireland for supporting the creation of the WoodScape dataset. We want to thank Ciarán Eising (Valeo) and Ravi Kiran (Navya) for providing a detailed review.

REFERENCES

[1] V. R. Kumar, S. A. Hiremath, S. Milz, C. Witt, C. Pinnard, S. Yogamani, and P. Mader, "Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving," *arXiv preprint arXiv:1910.04076*, 2019.

[2] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold, and C. Wohler, "Autonomous driving goes downtown," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 6, pp. 40–48, 1998.

[3] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 3828–3838.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

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

[6] C. Wang, J. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] V. Ravi Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Monocular fisheye camera depth estimation using sparse lidar supervision," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2853–2858.

[8] G. Sistu, I. Leang, S. Chennupati, S. Yogamani, C. Hughes, S. Milz, and S. Rawashdeh, "Neurall: Towards a unified visual perception model for automated driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 796–803.

[9] S. Chennupati, G. Sistu, S. Yogamani, and S. A Rawashdeh, "Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[10] S. Chennupati, G. Sistu., S. Yogamani., and S. Rawashdeh., "Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving," in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2019, pp. 645–652.

[11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] P. Yadati and A. M. Namboodiri, "Multiscale two-view stereo using convolutional neural networks for unrectified images," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2017, pp. 346–349.

[13] M. Yahiaoui, H. Rashed, L. Mariotti, G. Sistu, I. Clancy, L. Yahiaoui, V. R. Kumar, and S. Yogamani, "Fisheyemodnet: Moving object detection on surround-view cameras for autonomous driving," *arXiv preprint arXiv:1908.11789*, 2019.

[14] M. Uřičář, P. Křížek, G. Sistu, and S. Yogamani, "Soilingnet: Soiling detection on automotive surround-view cameras," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019.

[15] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "Cam-convs: camera-aware multi-scale convolutions for single-view depth," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 11 826–11 835.

[16] A. E. Conrady, "Decentred lens-systems," *Monthly notices of the royal astronomical society*, vol. 79, no. 5, pp. 384–390, 1919.

[17] C. Hughes, P. Denny, E. Jones, and M. Glavin, "Accuracy of fish-eye lens models," *Applied Optics*, vol. 49, no. 17, pp. 3338–3347, 2010.

[18] J. P. Barreto, "Unifying image plane liftings for central catadioptric and dioptric cameras," *Imaging Beyond the Pinhole Camera*, 2006.

[19] B. Khomutenko, G. Garcia, and P. Martinet, "An enhanced unified camera model," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 137–144, 2016.

[20] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 552–560.

[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[22] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[23] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[26] L. Zhou, J. Ye, M. Abello, S. Wang, and M. Kaess, "Unsupervised learning of monocular depth estimation with bundle adjustment, super-resolution and clip loss," *arXiv preprint arXiv:1812.03368*, 2018.

[27] Z. Arıcan and P. Frossard, "Dense depth estimation from omnidirectional images," 2009.

[28] A. Geiger, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 3354–3361.

[29] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5667–5675.

[30] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 36–53.

[31] 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.

[32] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *arXiv preprint arXiv:1810.06125*, 2018.

[33] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8001–8008.

[34] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.

[35] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9308–9318.

[36] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[37] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[39] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[40] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, 2019, pp. 9593–9604.

[41] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006. International Society for Optics and Photonics, 2019.

[42] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[43] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[44] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.