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

Varun Ravi Kumar1,4†, Senthil Yogamani2†, Markus Bach1, Christian Witt1, Stefan Milz3,4 and Patrick Mäder4
1 Valeo DAR Kronach, Germany 2 Valeo Vision Systems, Ireland 3 Spleenlab GmbH, Germany 4 Technische Universität Ilmenau, Germany 1 co-first authors

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.

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]. 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 [10]. 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. [11] 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 [12], and soiling detection [13] were demonstrated on fisheye images without rectification.

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 [14], 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

---

†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 [15] 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:

$$
x = x_c/s_c, \quad y = y_c/s_c
$$

$$
x' = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2)
$$

$$
y' = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy
$$

$$
i = f_x \cdot x' + c_x, \quad j = f_y \cdot y' + c_y
$$

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 $r(\theta)$ [16]. 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) [17] and eUCM (Enhanced UCM) [18]. 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 [19] 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: $r(\theta) = f \cdot \sin\theta/\cos\theta + \alpha\sin^2\theta + \beta\sin^2\theta\cos^2\theta - \cos^2\theta$
4) Rectilinear: $r(\theta) = f \cdot \tan\theta$
5) Stereographic: $r(\theta) = 2f \cdot \tan(\theta/2)$
6) Double Sphere: $r(\theta) = f \cdot \sin\theta/\sqrt{\sin^2\theta + (\xi + \cos\theta)^2 + (1 - \alpha)(\xi + \cos\theta)}$

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 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. For a horizontal FOV greater than 180°, there are rays incident from behind the camera, making it theoretically

![Fig. 2 Illustration of distortion correction in KITTI and WoodScape datasets.](image-url)
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 [21]. 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 and cropping to a standard resolution as per work on millions of cameras in commercial vehicles. Thus instance, dataset capture and training are typically performed when driving in a hot region. Thus a model that relies on resampling distortion, which is caused by interpolation errors at inference time, a small region is expanded to a larger one in the warped image. Particularly strong in the periphery of fisheye lenses because it is impossible to establish a complete mapping to a rectilinear image. Besides, the warping step is needed at inference time, which consumes significant computing power and memory bandwidth.

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:

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

(1)

Here, the pixel set of image \( I_t \) is represented by \( p_t \) and \( \Pi^{-1} \) 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 \( \Pi \) 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.

\[
\hat{p}_t = \Pi(T_{t \rightarrow t'} \Pi^{-1}(p_t, \hat{D}_t)), \quad \hat{I}_{t' \rightarrow t} = \left\{ \hat{I}_{t'}^{ij} \right\}
\]

(2)

Because the warped coordinates \( \hat{i}, \hat{j} \) are continuous, differentiable spatial transformer network [23] can be used to synthesize \( I_{t' \rightarrow t} \) by bilinear interpolation of the four pixels of \( I_t ' \) nearby \( \hat{p}_t \). The \( \left( \ldots \right) \) symbol denotes the corresponding operator used for sampling.

Reconstruction Loss The L1 pixel-wise loss is coupled with Structural Similarity (SSIM) [24], to form an image reconstruction loss \( L_r \) between the target image \( I_t \) and the reconstructed target image \( \hat{I}_{t' \rightarrow t} \) given by Eq. 3 below.

\[
\hat{L}_r(I_t, \hat{I}_{t' \rightarrow t}) = \omega \frac{1 - \text{SSIM}(I_t, \hat{I}_{t' \rightarrow t}, \mathcal{M}_{t' \rightarrow t'})}{2} + (1 - \omega) \frac{\| (I_t - \hat{I}_{t' \rightarrow t}) \odot \mathcal{M}_{t' \rightarrow t'} \|_1}{\| I_t \|_1}
\]

(3)

The binary mask \( \mathcal{M}_{t' \rightarrow t'} \) from [1] is incorporated, element-wise multiplication is denoted by \( \odot \) and \( \omega \) 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 95th percentile based on [25] 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 \( \sigma \) can be translated to depth with \( D = 1/(x \sigma + y) \) based on the fact that \( \text{depth} \propto 1/\text{disparity} \) 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 [26] for...
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], [28], [29].

\[
L_s(\hat{D}_t) = |\partial_i \hat{D}_t^* e^{-|\partial_i T_t|} + |\partial_j \hat{D}_t^* e^{-|\partial_j T_t|})
\] (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. \(\hat{D}_t^* = \hat{D}_t^{-1} / 1\), where \(\hat{D}_t\) 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 \(L_r\) applied on forward and backward sequences and an edge-aware smoothness term \(L_s\) to regularize depth. Additionally, \(L_{dc}\), 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 [23], 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 of pixels, scales and batches.
\[
\mathcal{L} = \sum_{n=1}^{4} \frac{\mathcal{L}_n}{2^{n-1}},
\]
\[
\mathcal{L}_n = \alpha_n \mathcal{L}_f + \beta_n \mathcal{L}_d + \gamma_n \mathcal{L}_c + \beta_n \mathcal{L}_s.
\]

IV. IMPLEMENTATION DETAILS

The depth estimation network is mainly based on FisheyeDistanceNet [1]. We use Pytorch [37] and employ Ranger (RAdam [38] + LookAhead [39]) 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 \(4 \times 10^{-4}\) with OneCycleScheduler [40]. The network’s sigmoid output \(\sigma\) is converted to depth with \(D = \frac{1}{(\alpha \cdot x + \gamma)}\) 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 \(\times \) 224 pixels, and for WoodScape fisheye images 512 \(\times \) 256 pixels as the network input to maintain the original aspect ratio. The loss weighting factors \(\beta\) and \(\gamma\) of smoothness and cross-sequence depth consistency loss are set to 0.001. To remove checkerboard artifacts in the sub-pixel convolution [41], the final convolutional layers are initialized in a particular manner before the pixel shuffle operation as described in [36].

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. [35]. 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 [33]. The WoodScape [34] 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. [30]. 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. 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], [42]; (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 [43]; (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.
[42] 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.

[43] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” Distill, vol. 1, no. 10, p. e3, 2016.