Acta Univ. Sapientiae, Informatica, **16, 1** (2024) 62–77

DOI: 10.2478/ausi-2024-0005

Integrating Optical Flow into Deep Learning based Distortion Correction

Szabolcs-Botond LŐRINCZ-MOLNÁR
Independent Researcher
✉ lorincz.szabolcs.botond@gmail.com
0000-0002-2202-9491

Szabolcs PÁVEL
Babeş-Bolyai University
✉ szabolcs.pavel@ubbcluj.ro
0000-0002-8825-2768

Abstract. One crucial task in 3D computer vision is the correction of geometric distortions, since most algorithms rely on the assumption that the image formation process can be described by a specific camera model, e.g. the pinhole camera model. In an autonomous driving scenario, however, the front-facing camera is most commonly placed behind the windshield, causing complex, nonlinear distortions.

Previous attempts have been made to undistort such images using deep learning based methods. The input of these deep networks usually consists of one or more images, and they optionally include additional tasks such as semantic segmentation to improve the results.

We hypothesize that the well-constrained nature of optical flow in rigid, static scenes provides useful cues for the process of image undistortion. By using optical flow as an additional input, we present a multi-view distortion correction method achieving superior results on both synthetic and real-world images compared to previous works, demonstrating the usability of optical flow for correcting highly complex distortions.

Key words and phrases: camera calibration, distortion correction, deep learning

1 Introduction

Front-facing cameras in autonomous driving systems are most commonly placed behind the vehicles' windshields, therefore the captured image sequences suffer from

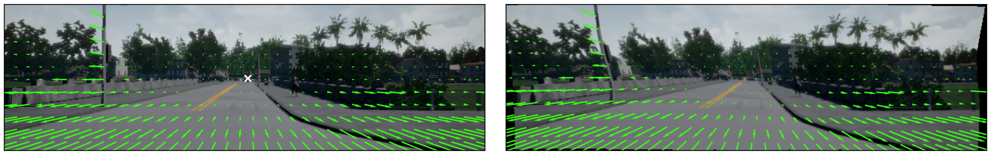

Figure 1: In the case of a calibrated camera and pure forward motion (left) all optical flow vectors emerge from the focus of expansion (white cross), situated at the principal point. In the case of geometric distortions (right), aberrations are induced in both the magnitude and direction of the vectors. The irregularity of distorted optical flow vectors provides useful cues for distortion correction.

complex geometric distortions caused by light refraction. This anomaly impacts the performance of various computer vision pipelines, inducing errors for instance in scene reconstruction, depth estimation and in camera based driver assistance systems in general. Such kind of errors are not admissible in critical systems, thus, these distortions must be corrected.

Several attempts have been made to correct distortions caused by different refractive surfaces, *e.g.* radial distortions caused by the camera lens [1] or tangential distortions, the root of which lies at the nonparallelism of the lens and the image plane. Distortions caused by wide angle (fisheye) lenses typically used in autonomous driving systems have also been succesfully corrected [2].

Recent experiments show that the correction of more complex distortions, such as the ones caused by windshields is also feasible, by employing deep neural networks and using additional tasks to guide the process of undistortion, such as semantic segmentation [3].

In this work, we extend previously proposed methods for correcting complex distortions caused by car windshields and achieve superior performance on both synthetic and real-world data sets. We also show experimentally that using optical flow as an additional input to a deep learning based distortion correction method improves the performance of the system by exploiting the predictability and regularity of optical flow vector directions and magnitudes. The proposed method can be trained without the need of conducting rigorous measurements to obtain ground truth distortion fields, by leveraging differentiable image sampling, enabling us to produce the undistorted image jointly with estimating the distortion parameters.

2 Related Work

Calibration The task of predicting distortion parameters and correcting distortions has been studied for a few years by now. Early methods tried to correct simpler distortions such as radial or tangential distortions, the former being caused by the camera lens, the later by the lens being non parallel relative to the image plane. These methods can be categorized into two major groups: self-calibration methods exploiting geometric constraints based on multiple view image sequences [4]–[6], and static calibration approaches using a calibration object or pattern [7]–[11] where relative positions of specific keypoints are known, but their perceived positions are distorted during projection.

Deep Learning based Distortion Correction Recent distortion correction methods started to employ deep learning, specifically convolutional neural networks (CNNs) to predict radial distortion parameters based on arbitrary single view input images [1], without the need of having a calibration pattern. Later, CNNs have been used to estimate the parameters of more complex distortions, such as fisheye distortions caused by wide angle cameras [2] or distortions caused by windshields [3], using semantic segmentation as an additional task to guide the correction process.

Distortion Model Each distortion estimation or correction algorithm employs a specific distortion model, ranging from simple models *e.g.* Brown's polynomial model [12] for radial and tangential distortions to the more complex methods introduced in [13]–[15], explicitly modeling the refractive surface. In this work, we use thin plate spline (TPS) interpolation [16] to model the two dimensional geometric distortions caused by windshields.

Optical Flow based Image Reconstruction Studies of using optical flow in the distortion correction process have also been conducted. Non-rigid geometric distortions caused by atmospheric turbulences were corrected in [17] using an optical flow scheme and a non local total variation (TV) regularization, while in [18] distortions of the same kind are corrected using a non-rigid image registration algorithm based on B-splines, embedded in a Bayesian framework with bilateral TV regularization. Both methods focus on restoring images using video sequences having no camera motion, assuming a constant scene. In an autonomous driving scenario this assumption does not hold, therefore a different approach must be taken.

Supervision through View Synthesis Several methods rely on direct supervision from ground truth distortion parameters [1], [19]. In the case of simple forms of distortions it is possible to obtain these parameters for specific settings separately. However, the diversity of windshields and the complexity of the distortion caused by them makes the collection of large data sets a tedious task. The introduction of differentiable inverse warping by [20] led to the emergence of several methods employing novel view synthesis [21] as supervision to solve the problem of layered 3D scene representation [22], unsupervised learning of depth, ego-motion [23] and optical flow [24].

In this work we employ a reconstruction loss as supervision based on Multi Scale Structural Similarity [25]. In contrast with novel view synthesis based methods, instead of synthesizing a new image from a different view given a single image, we generate corrected images given three distorted images and two corresponding optical flow maps. This enables the method to be trained even in the case of real-world, complex and highly variable distortions, when the estimation of ground truth parameters is not feasible, but distorted and correct image pairs can be obtained.

3 Motion Field Constraints

Our main hypothesis is that using optical flow as an additional input to a distortion correction network provides important information about the distortion field, and as a result can improve the performance of the deep learning system. In an ideal scenario the optical flow corresponds to the motion field – the projection of 3D velocity vectors to the image plane [26], [27]. Assuming a static, rigid scene the 3D velocity vectors are only influenced by the ego-motion of the camera. In a front-facing camera used in autonomous driving scenarios further assumptions can be made, such that the dominant motion component is the forward translation, and in some cases the yaw rotation. These assumptions cause the optical flow to become well predictable, and deviations from the predicted optical flow field are in part caused by the geometric distortions of the imaging system.

Let $\boldsymbol{P} = (X, Y, Z)$ be a 3D point in the scene, with a corresponding velocity vector (time derivative of $\boldsymbol{P}$) $\boldsymbol{V} = (V_x, V_y, V_z)$. The perspective projection of $\boldsymbol{P}$ to the image plane is denoted by $\boldsymbol{p} = (x, y)$, and is given by the first two components of the vector $\frac{f\boldsymbol{P}}{Z}$, where f denotes the focal distance of the camera. Then the 2D velocity vectors $\boldsymbol{v} = (v_x, v_y)$ of the motion field can be computed as a function of the 3D position $\boldsymbol{P}$ and velocity $\boldsymbol{V}$ by differentiating the 2D pose $\boldsymbol{p}$ w.r.t. the time, resulting in:

$$v_x = \frac{fV_x - xV_z}{Z} \qquad v_y = \frac{fV_y - yV_z}{Z}. \tag{1}$$

The 3D velocity vector can be written as $\boldsymbol{V} = \boldsymbol{T} + \boldsymbol{\Omega} \times \boldsymbol{P}$, where $\boldsymbol{T} = (T_x, T_y, T_z)$ is a linear velocity (translation), and $\boldsymbol{\Omega}$ is the angular velocity. Assuming zero angular velocity, the velocity vector is equal to the translation $\boldsymbol{T}$ of the 3D points, and it is independent of the 3D position of the point. As a result, Eq. (1) is reduced to:

$$v_x = \frac{fT_x - xT_z}{Z} \qquad v_y = \frac{fT_y - yT_z}{Z}. \tag{2}$$

In this case we can observe, that the motion field is composed of radial vectors emerging from a common point on the image called the focus of expansion (in the case of forward motion). The focus of expansion is influenced by the T_x and T_y components of the translation. If both x and y components are zero ($\boldsymbol{T} = (0, 0, T_z)$), i.e. we have pure forward motion, the focus of expansion corresponds to the principal point of the image. One more desirable property is that the scene structure (the depth Z of the 3D points) only influences the magnitude, but not the direction of the velocity vectors. As a consequence the direction of these vectors by themselves can provide useful constraints, while the large changes in magnitude can signal object boundaries or occlusions. An example of the motion field with pure forward motion can be seen in Fig. 1, both in the case of a calibrated camera and in the presence of complex distortions caused by a windshield.

One important limitation of optical flow based distortion estimation in forward motion scenarios is that optical flow provides no information about the distortions in the radial direction [19]. An ambiguity exists where scene depth and radial distortions both influence only the magnitude of the motion field vectors, and an infinite number of depth - distortion pairs can result in the same motion field. In fact, this is a major factor in making optical flow based distortion estimation using classic computer vision challenging. A learning-based system however, despite this ambiguity, is still able to filter out the information relevant for the given task, therefore optical flow remains a useful input for distortion estimation.

4 Methods

4.1 Distortion Model

The distortion model in this work is identical to the one proposed in [3] to provide a fair comparison between the two methods and to allow us to properly quantify the effects of integrating optical flow into the distortion correction process.

The model relies on a pair of *thin plate splines (TPS)* forming a two dimensional linear map. The TPS transformation $\boldsymbol{f}_{tps}$ consists of two parts, the first being an affine transformation, while the second corresponding to the superposition of geometrically independent affine-free deformations [16], and is given by

$$\boldsymbol{f}_{tps}(G_i) = A \begin{bmatrix} G_i \\ 1 \end{bmatrix} + \sum_{k=1}^{n} \varphi(\left\|\boldsymbol{p}'_k - G_i\right\|_2) \cdot \boldsymbol{w}_k, \tag{3}$$

Table 1: Mean and Standard Deviation of Original Distortion Vector Norms

Data Set	Mean (px)	SD (px)
DC Test	8.46	3.92
DK 00	8.59	3.32

where $G_i = [x_i, y_i]^\top$ represents the image coordinates on the undistorted target image and n corresponds to the number of control points, in our case $n = 16$. The TPS kernel is denoted by $\varphi(r) = r^2 log(r)$, where r represents the L^2 distance between two points, with $P' = [\boldsymbol{p}'_1, \boldsymbol{p}'_2, \ldots, \boldsymbol{p}'_n] \in \mathbb{R}^{2\times n}$ being the coordinates of target control points. In our case points P' are evenly distributed and fixed on a 4×4 grid, whereas the coordinates of source control points $P = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n] \in \mathbb{R}^{2\times n}$ have to be estimated based on the distorted images and optical flows. The sampling grid is obtained by interpolating the displacements between point correspondences in P' and P. The transformation can be efficiently implemented by matrix operations as detailed in [3].

The properties of the map enables it to model various types of complex two dimensional deformations such as skeletal shape abnormalities caused by Apert syndrome [16], or even geometric distortions caused by refractive surfaces [3].

In this work, we applied the same parametric distortions sampled from a distribution derived from real-world measurements in the presence of windshields as in [3]. The mean and standard deviation of distortion norms reported in Table 1 is expressed in pixels in the distorted images.

4.2 Proposed Solution

In order to solve the problem of geometric distortion correction, we propose an end-to-end architecture similar to the architecture presented in [3] with some modifications.

First, the inputs of the network are three consequent RGB images, making our approach a member of multi-view distortion correction methods. As a direct consequence, the outputs of the network are also three consequent, corrected images, in addition to the estimated distortion parameters.

Furthermore, we feed two optical flows corresponding to the three images for guiding the process of distortion correction instead of employing an additional task

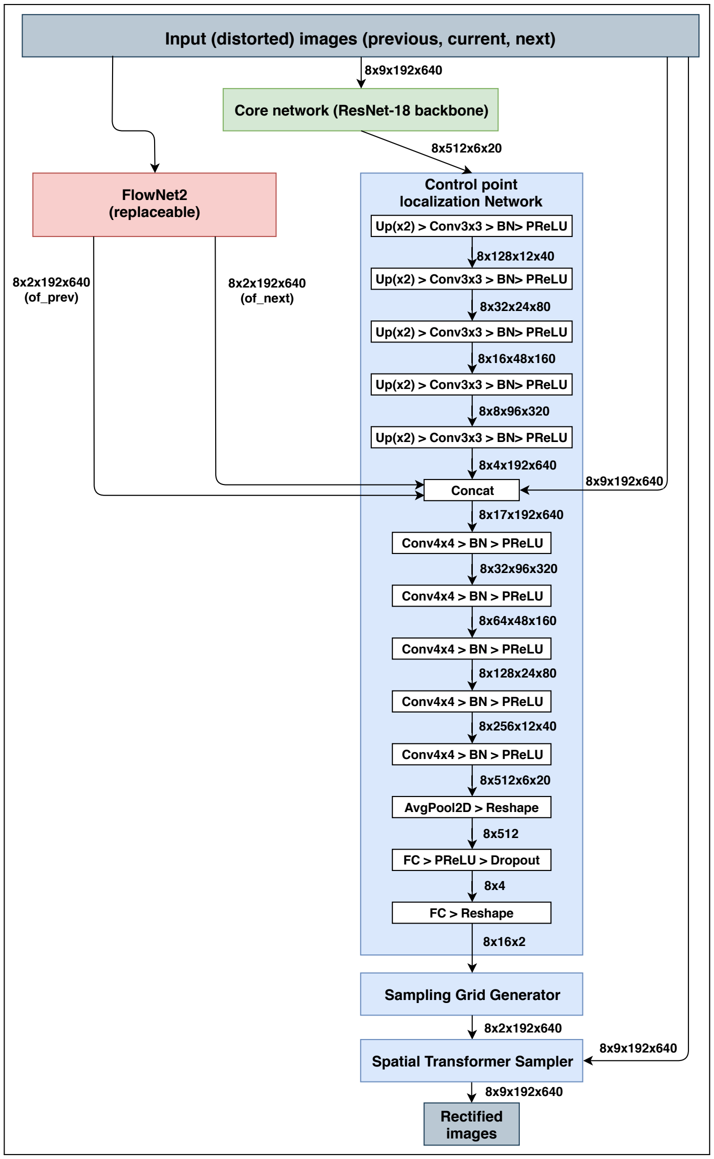

Figure 2: The proposed architecture consists of three key modules: the core network (green), the optical flow network (red) and a Spatial Transformer module (blue). The input of the network is formed of three consequent distorted images, based on which it produces two optical flows. The optical flow maps are concatenated to the input images and the upsampled features generated by the core network and they are jointly used for estimating the parameters of the TPS transformation to correct the images.

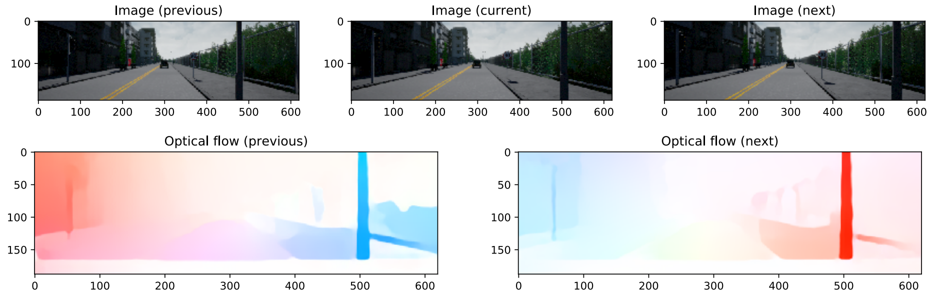

Figure 3: Using three consequent image frames (top), we calculate the optical flow between the second- and the first, and between the second- and the third image (bottom) using FlowNet 2.0, which we later use for distortion correction.

such as semantic segmentation as in [3].

The model corrects images in two steps: a feature extraction step, and a distortion correction step. For feature extraction we use ResNet-18 [28] pre-trained on ImageNet [29] as the core network. To obtain optical flow based on the three consequent input images, we use the pre-trained PyTorch [30] implementation [31] of FlowNet 2.0 [32].

For distortion correction we utilize the Spatial Transformer module [20], a differentiable image warping mechanism, also integrating the TPS interpolation based distortion model into the framework.

The Spatial Transformer module first estimates the parameters of the TPS transformation, being the coordinates of the source control points. Then, a sampling grid is generated based on the source and target control point coordinates, in our case this corresponds to the inverse of the distortion field. Lastly, the sampling grid is used to sample the distorted images, then the pixel values in the corrected images are calculated by bilinear interpolation. Since all three steps are differentiable, end-to-end learning is achievable. The detailed architecture is presented in Fig. 2.

In our experiments we explore the possibility of minimizing multiple loss functions separately and in a joint fashion. In order to enforce the reconstruction of the corrected images in terms of luminance, contrast and structure, the reconstruction loss $\mathcal{L}_r$ proposed in [3] given by Eq. (4) is used which is based on Multiscale Structural Similarity (MS-SSIM) [25] between a ground truth correct image (I) and the predicted corrected image ($\hat{I}$).

$$\mathcal{L}_r = -\frac{\text{MS-SSIM}(I, \hat{I}) + 1}{2} \tag{4}$$

By minimizing the reconstruction loss, the network is trainable even on data sets containing real-world distortions, when the ground truth sampling grid is hard or even impossible to obtain accurately, but distorted and correct image pairs are available.

In our experiments synthetic distortions are applied, thus, the ground truth sampling grid can be determined. For this reason, we experiment with direct minimization of the mean squared error (MSE) between the ground truth sampling grid and the predicted sampling grid, namely the grid loss $\mathcal{L}_g$ proposed in [3]. We also quantify its effect on model performance.

The joint loss is a linear combination of the two losses given by the following equation:

$$\mathcal{L} = \mathcal{L}_r + \lambda_1 \mathcal{L}_g, \tag{5}$$

where λ_1 is a weighting coefficient balancing the two loss functions set to $\lambda_1 = 100$ experimentally.

4.3 Experimental Setup

With the purpose of being able to correct not only synthetic but distorted real-world images also, we followed the experimental setups of Lőrincz et al. [3] and constructed two data sets, which we name Distorted Carla (DC) and Distorted KITTI (DK).

To construct the DC data set, 10000 images are generated using Carla driving simulator [33] with the same settings as in [3]. DK data set contains images from KITTI odometry data set [34] consisting of real-world sequences captured in Karlsruhe, Germany. In our experiments the first seven sequences of KITTI odometry data set are utilized (ranging from 00 to 06), specifically images captured by the left camera, which means a total of 15223 images.

Since we use optical flow merely as an additional input rather than as an additional task for guiding distortion correction, for both data sets we generate in advance two optical flows with FlowNet 2.0 based on every three consequent images. The first optical flow corresponds to the optical flow between the second and first frames, while the second optical flow is generated based on the second and the third frame of a sequence, as demonstrated in Fig. 3, resulting in 9998 optical flow pairs in the case of DC data set and 15221 pairs in the case of DK data set.

We follow the same data set splitting and model training procedure as in [3]. The training of the networks is conducted on DC Train (8,000 images) for a total of 10 epochs, testing is conducted on both DC Test (2,000 images) and on sequence 00 of DK data set (4,539 images). The trained networks are further fine-tuned on

Table 2: Mean and Standard Deviation of Residual Distortion Vector Norms

Method	Fine-Tune	Test	$\mathcal{L}_r$	$\mathcal{L}_g$	Sem. Seg.	Opt. Flow	Mean (px)	SD (px)
Lőrincz et al. [3]	✗	DC Test	✓				2.26	1.49
				✓			2.25	1.59
			✓	✓			2.15	1.47
			✓		✓		1.98	1.40
				✓	✓		2.15	1.53
			✓	✓	✓		2.06	1.45
Ours			✓			✓	1.72	1.07
				✓		✓	1.52	0.62
			✓	✓		✓	1.43	0.71
Lőrincz et al. [3]	✗	DK 00	✓				1.75	1.10
				✓			2.28	1.52
			✓	✓			1.99	1.35
			✓		✓		1.65	1.09
				✓	✓		1.37	0.88
			✓	✓	✓		2.53	1.31
Ours			✓			✓	4.99	1.75
				✓		✓	2.65	1.09
			✓	✓		✓	3.47	1.18
Lőrincz et al. [3]	DK 01-06	DK 00	✓				1.24	0.72
				✓			1.33	0.67
			✓	✓			1.30	0.70
			✓		✓		1.30	0.69
				✓	✓		1.22	0.72
			✓	✓	✓		1.24	0.70
Ours			✓			✓	1.16	0.75
				✓		✓	1.06	0.71
			✓	✓		✓	1.06	0.68

sequences ranging from 01 to 06 of DK data set (10,684 images) for another 10 epochs and are also evaluated on sequence 00.

In our experiments we compared the performance of the proposed distortion correction network with the method presented in [3]. We also performed an ablation study to quantify the individual effects of integrating optical flow into the distortion correction process. Additionally, we examined the shortcomings of the proposed method and discussed further directions worth discovering to improve the current results.

5 Results

In Table 1 we present the mean and standard deviation of distortion vector norms in the two data sets providing a basis of comparison for estimating the performance of the variants of the proposed distortion correction method. The performance comparison of our solution and the method proposed by [3] is presented in Table 2 in terms of mean and standard deviation of residual distortion vector norms measured in pixels.

The effect of integrating optical flow into the distortion correction process is also quantified in Table 2 in all different settings: with- and without fine-tuning on DK data set and by minimizing separately or jointly the reconstruction and grid loss functions. Overall, one can see that our proposed distortion correction method using optical flow as an additional input performs the best in almost all cases. The best performing optical flow based model on DC Test reduces distortion vector norms to 1.43 ± 0.71 pixels, while on DK 00 this value is equal to 2.65 ± 1.09 pixels without fine-tuning and 1.06 ± 0.68 pixels with fine-tuning.

Similarly to the segmentation based system proposed by [3], we investigated the performance of the model in the case in which only the distorted and correct images are known, and we do not make use of the ground truth sampling grid during training, which is considered to be unknown (similar to the case of real-world distortions). In this case, the optical flow based model achieves 1.72 ± 1.07 pixels on DC Test, 4.99 ± 1.75 pixels on DK 00 without fine-tuning, and 1.16 ± 0.75 pixels with fine-tuning. One can see, that the model achieves comparable results without direct supervision from ground truth sampling grids, demonstrating the applicability of the network in the case of real-world distortions.

6 Discussion

The only exceptions where the optical flow based model does not outperform the method proposed in [3] are the tests conducted on real world images (DK 00) without fine-tuning the network on real-world images (sequences 01–06 of DK).

This phenomena is explained by the optical flow network used in our experiments producing substantially different optical flows based on synthetic images compared to the real-world images. Thus, our method needs to refine its distortion parameter estimations by fine-tuning it on the real-world images and corresponding optical flows.

Our experiments do not address certain possibilities, therefore further potential extensions need to be mentioned. First, the optical flow based distortion correction method processes image sequences, which have to be captured from multiple views, consequently, the camera has to be in motion in order to have optical flow vectors suitable for distortion correction.

In this case, it is possible to exploit the constant nature of the distortion caused by refractive surfaces by estimating the distortion parameters using images captured in adequate scenes, when optical flow vector norms are above a certain threshold, then calculating the mean sampling grid. Based on the calculated sampling grid, each new image can be sampled to correct the geometric distortions, even if the camera is not in motion.

Further experiments are also needed to achieve robustness regarding the method used for generating optical flow. Our results show that currently the proposed method is sensitive to the quality of the produced optical flow to a certain degree, without fine-tuning on real-world images its performance is inferior to the method proposed in [3]. This could be avoided by training with various optical flow methods, injecting variety in the data set. This was out of scope of our current experiments, however.

Another potential improvement would be achieved by extending the scope of this paper to real-world distortions by collecting real-world distorted and undistorted image pairs, enabling us to train and test the system in the presence of real-world distortions, in contrast to the data sets used in this work, which only contain images on which synthetic distortions were applied.

7 Conclusion

In this work we presented a deep learning based distortion correction method which is capable of correcting a wider range of geometric distortions compared to existing methods, based on three consequent RGB images using two corresponding optical flows as additional inputs for guiding the process of distortion correction.

We also showed that the predictability and regularity of optical flow vector directions typical to autonomous driving scenarios can be exploited to assist the distortion correction method.

Our experimental results proved the hypothesis, that using optical flow as an additional input enhances the distortion correction method compared to employing an additional task such as semantic segmentation introduced in [3].

We detailed the disadvantages and constraints of the proposed system as well, and proposed solutions for each potential failure case. Addressing the mentioned problems would be the most important direction of development.

Data Availability: This work uses data sets derived from the public KITTI odometry dataset [34] and a data set generated using the open-source CARLA simulator [33]. To generate the distorted images, proprietary windshield distortion measurement data was used, and as a consequence the final derived data set can not be published.

References

[1] J. Rong, S. Huang, Z. Shang, and X. Ying, "Radial lens distortion correction using convolutional neural networks trained with synthesized images," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., Cham: Springer International Publishing, 2017, pp. 35–49, ISBN: 978-3-319-54187-7 (⇒ 63, 64).

[2] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao, "Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification," *arXiv preprint arXiv:1804.04784*, 2018 (⇒ 63, 64).

[3] S.-B. Lőrincz, S. Pável, and L. Csató, "Single view distortion correction using semantic guidance," in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–6 (⇒ 63, 64, 66, 67, 69–74).

[4] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, IEEE, vol. 1, 2001, pp. I–I (⇒ 64).

[5] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, 2007 (⇒ 64).

[6] G. P. Stein, "Lens distortion calibration using point correspondences," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, IEEE, 1997, pp. 602–608 (⇒ 64).

[7] C. Bräuer-Burchardt and K. Voss, "Automatic lens distortion calibration using single views," in *Mustererkennung 2000*, Springer, 2000, pp. 187–194 (⇒ 64).

[8] B. Prescott and G. McLean, "Line-based correction of radial lens distortion," *Graphical Models and Image Processing*, vol. 59, no. 1, pp. 39–47, 1997 (⇒ 64).

[9] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987 (⇒ 64).

[10] A. Wang, T. Qiu, and L. Shao, "A simple method of radial distortion correction with centre of distortion estimation," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 3, pp. 165–172, 2009 (⇒ 64).

[11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000 (⇒ 64).

[12] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering and Remote Sensing*, 1966 (⇒ 64).

[13] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari, "A theory of multi-layer flat refractive geometry," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3346–3353 (⇒ 64).

[14] S. Morinaka, F. Sakaue, J. Sato, K. Ishimaru, and N. Kawasaki, "3d reconstruction under light ray distortion from parametric focal cameras," *Pattern Recognition Letters*, vol. 124, pp. 91–99, 2019 (⇒ 64).

[15] S. Pável, C. Sándor, and L. Csató, "Distortion estimation through explicit modeling of the refractive surface," in *International Conference on Artificial Neural Networks*, Springer, 2019, pp. 17–28 (⇒ 64).

[16] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, pp. 567–585, 1989 (⇒ 64, 66, 67).

[17] Y. Mao and J. Gilles, "Non rigid geometric distortions correction-application to atmospheric turbulence stabilization," *Inverse Problems & Imaging*, vol. 6, no. 3, p. 531, 2012 (⇒ 64).

[18] X. Zhu and P. Milanfar, "Image reconstruction from videos distorted by atmospheric turbulence," in *Visual Information Processing and Communication*, International Society for Optics and Photonics, vol. 7543, 2010, 75430S (⇒ 64).

[19] B. Zhuang, Q.-H. Tran, G. H. Lee, L. F. Cheong, and M. Chandraker, "Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 3766–3773 (⇒ 64, 66).

[20] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025 (⇒ 65, 69).

[21] R. Szeliski, "Prediction error as a quality metric for motion and stereo," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, vol. 2, 1999, pp. 781–788 (⇒ 65).

[22] S. Tulsiani, R. Tucker, and N. Snavely, "Layer-structured 3d scene inference via view synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 302–317 (⇒ 65).

[23] 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 (⇒ 65).

[24] 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 (⇒ 65).

[25] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Asilomar Conference on Signals, Systems & Computers*, vol. 2, 2003, pp. 1398–1402 (⇒ 65, 69).

[26] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 208, no. 1173, pp. 385–397, 1980 (⇒ 65).

[27] A. Distante and C. Distante, *Handbook of Image Processing and Computer Vision: Volume 3: From Pattern to Object*. Springer Nature, 2020 (⇒ 65).

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016 (⇒ 69).

[29] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y (⇒ 69).

[30] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in pytorch," 2017 (⇒ 69).

[31] F. Reda, R. Pottorff, J. Barker, and B. Catanzaro, *Flownet2-pytorch: Pytorch implementation of flownet 2.0: Evolution of optical flow estimation with deep networks*, `https://github.com/NVIDIA/flownet2-pytorch`, 2017 (⇒ 69).

[32] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. [Online]. Available: `http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17` (⇒ 69).

[33] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16 (⇒ 70, 74).

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012 (⇒ 70, 74).

Received: 19.06.2024; *Revised: 30.06.2024*; *Accepted: 07.07.2024*