Radial Lens Distortion Correction by Adding a Weight Layer with Inverted Foveal Models to Convolutional Neural Networks

Yongjie Shi, Danfeng Zhang, Jingsi Wen, Xin Tong, Xianghua Ying and Hongbin Zha
Key Lab of Machine Perception (MOE)
School of Electronics Engineering and Computer Science
Peking University

***Abstract*—Radial lens distortion often exists in images taken by commercial cameras, which does not satisfy the assumption of pinhole camera model. Eliminating the radial lens distortion of an image is necessary as a preprocessing step for many vision applications. Some paper has employed Convolutional Neural Networks (CNNs), to achieve radial distortion correction. They generated images with a large number of images of high variation of radial distortion, which can be well exploited by deep CNN with a high learning capacity, and reach the state-of-the-art results. In this paper, we claim that a weight layer with inverted foveal models can be added to these existing CNNs methods for radial distortion correction. In the widely used very deep Resnet-18 model, our method achieves about 20 percent decrease in the loss function with faster convergence compared to the previous methods.**

I. Introduction

Classical computer vision algorithms, such as structure from motion and single view metrology, usually depend critically on the assumption of the ideal pinhole camera model [8]. Unfortunately, most lenses in commercial and industrial cameras often inherently have lens distortion, and the radial distortion is the most critical especially in wide angle lenses [12], [13], [19], [20], [21]. Some different kinds of distortion models [2], [4], [14], [18] are presented to characterize the radial distortion. The division model introduced by Fitzgibbon [7] is one of the most popular one. So many straight lines exist in structured scenes, and the lines would be projected to curves in the image plane because of radial distortion [6]. As we know, straight line should be projected into straight line under pinhole model. Several papers for estimating radial distortion have investigated into the so-called plumb line idea based on such fact [3], [10], [16], [1].

Recently, Rong et al. [15] employ CNNs, to achieve radial distortion correction. They augmented real images by synthesizing thousands of distorted images which cover a high diversity in distortion coefficient. The CNN-based distortion coefficient estimator significantly outperformed the state-of-the-art method, tested on real image dataset. The aim of this research is to demonstrate that it is possible to use inherent lens aberrations in the images for the detection of radial distortion coefficients. Inconsistency in the degree of lens distortion leads to different appearances of line images in different portions of the image. The lens radial distortion causes non-linear geometrical distortion on images. For the central part of an image with radial lenses distortion, the distortions are usually too small to be caught. However, for the portions near the boundary of a distorted image, the distortions may be very serious. To infer the distortion coefficients from a distorted image, it seems that the portion near the image boundary may provide more contributions than the central portion of the image. Therefore, we introduce a weight layer with inverted foveal models into the existing CNNs methods, which makes different portions of an image have their corresponding weights. The performances of methods with different weight layers are shown in Fig. 1.

II. Algorithm

Our total framework can be seen in Fig. 2. Firstly, we use the division distortion model of radial distortion to synthesize the distorted image by using real images in ImageNet [5], and then we add an inverted foveal weight layers to CNN and use the synthetic distorted image to train it. Given a real distorted image, the network can be applied to predict the distortion coefficient.

A. Division Distortion Model of Radial Distortion

There are many radial lens distortion correction methods in literature. Usually, we assume the distortion center is already known. The distance from the original image center to the image point is defined as the distorted radius r_d, and the distance corresponding to the undistorted image point is defined as the undistorted radius r_u. For a point in the distorted image $x_d = [u_d, v_d, 1]$, and its corresponding undistorted image point $x_d = [u_u, v_u, 1]$, we have

$$\begin{bmatrix} u_u \\ v_u \\ 1 \end{bmatrix} \propto \begin{bmatrix} u_d \\ v_d \\ \mu \end{bmatrix} \tag{1}$$

where μ is some unknow non-zero scale. We rewrite (1) as,

$$\begin{bmatrix} u_u \\ v_u \end{bmatrix} = \frac{1}{\mu} \begin{bmatrix} u_d \\ v_d \end{bmatrix} \tag{2}$$

and

$$r_u = \sqrt{u_u^2 + v_u^2} = \frac{1}{\mu} r_d = \frac{1}{\mu}\sqrt{u_d^2 + v_d^2}. \tag{3}$$

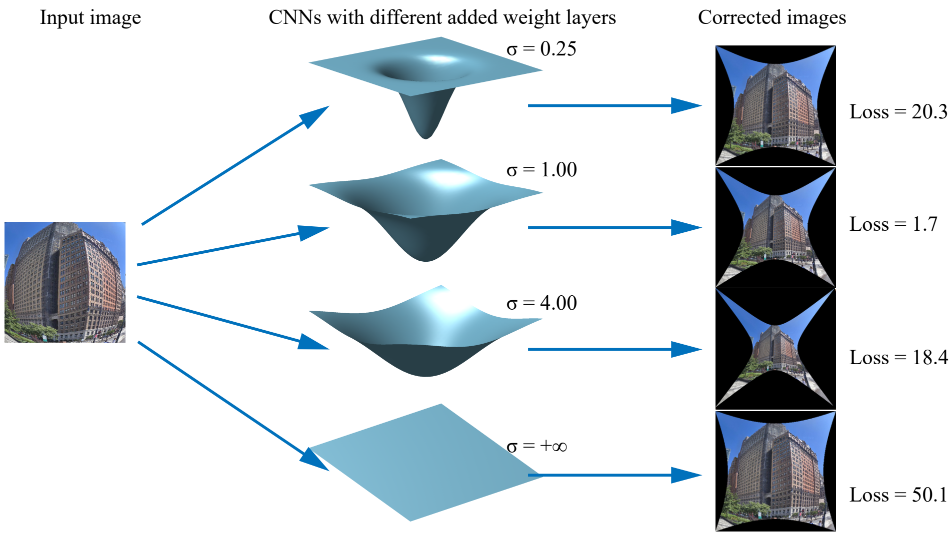

Fig. 1. Adding a weight layer of inverted foveal models with different σ into the existing CNNs methods. Note that, the method with $\sigma = 1.00$, is the best one with the smallest loss. The method with $\sigma = +\infty$, is corresponding to the case where no layer is added, i.e. the previous method [15].

The distortion model is described as

$$r_u = \frac{r_d}{1 + \xi_1 r_d^2 + \xi_2 r_d^4 + \xi_3 r_d^6 + \dots} \tag{4}$$

where $\xi_1, \xi_2, \xi_3, \dots$ represent the distortion coefficients. In the one-parameter distortion model proposed by Fitzgibbon [4], the relation between r_d and r_u is:

$$r_u = \frac{r_d}{1 + \xi r_d^2} \tag{5}$$

Therefore, we have

$$\begin{bmatrix} u_u \\ v_u \end{bmatrix} = \frac{r_u}{r_d} \begin{bmatrix} u_d \\ v_d \end{bmatrix} = \frac{1}{1 + \xi r_d^2} \begin{bmatrix} u_d \\ v_d \end{bmatrix} \tag{6}$$

When the value of distortion coefficients varies, the distortion varies accordingly. The more the absolute value of ξ gets, the stronger radial distortion the image will presents. We can easily observe that barrel distortion typically will have a negative term from ξ, while as for pincushion distortion, the value of ξ will be positive. For a real lens, however, the distortion coefficient ξ is usually negative, thus we only focus on the negative term for the distortion coefficient ξ.

B. Data Synthesis

It is hard to collect so many distorted images with ground truth. In this paper, we synthesize the distorted image using (6). Firstly, we choose some images in ImageNet [5], then we synthesize distorted images with its according distortion coefficient ξ we labeled. We have noted that, in application of radial lens, the magnitude of coefficient ξ is usually vary from 10^{-8} to 10^{-6}. From this view, we can randomly choose a set of numbers which are uniformly distributed on the interval from $-4 \times 10^{-6}$to -0.01×10^{-6} , this range is usually common radial lens distortion coefficients vary in. Therefore, we can simulate radial lens distortion using (6) and label reshaped images with generated distortion coefficients. Totally, we have 110000 synthetic distortion images with different distortion coefficient in train datasets. Test datasets include 12000 distortion images. The resolution of all images is 256×256.

C. Network Structures

CNNs have shown excellent performance in many computer vision and machine learning problems. Typical CNNs include Alexnet [11], GoogleNet [17], Resnet [9] et al. For our task, we adopt Alexnet and Resnet-18 architecture, which was designed for object recognition as part of the ImageNet challenge [5]. Alexnet consists of five convolutional layers, each followed by a non-linearity (rectified linear unit), and occasionally interspersed with pooling and local response normalization. This is followed by three fully connected layers. Unlike Alexnet, Resnet-18 has multiple basic blocks that are serially connected to each other, and there are also shortcut connections parallel to each basic block and it gets added to its output.

Using Alexnet and Resnet-18 network architecture, combined with (6), we can directly predict the distortion coefficient. Given the distorted image and the corresponding distortion coefficient, we can train the network, and predict the distortion coefficient in test stage [15]. However, this kind of way does not combine the distorted image properties. Distorted image have the center symmetry characteristics, and texture far from the image center has strong distortion. For predicting the distortion coefficient task, it is obviously that the farther the point from image center is, the more important it is. So we can manually increase the weight of the image points far from the image center, and reduce the weight of image points near it.

Therefore, we add a weight layer which can increase the weight of the point far from image center. Then fix this weight layer and update parameter of other layers in train stage. In

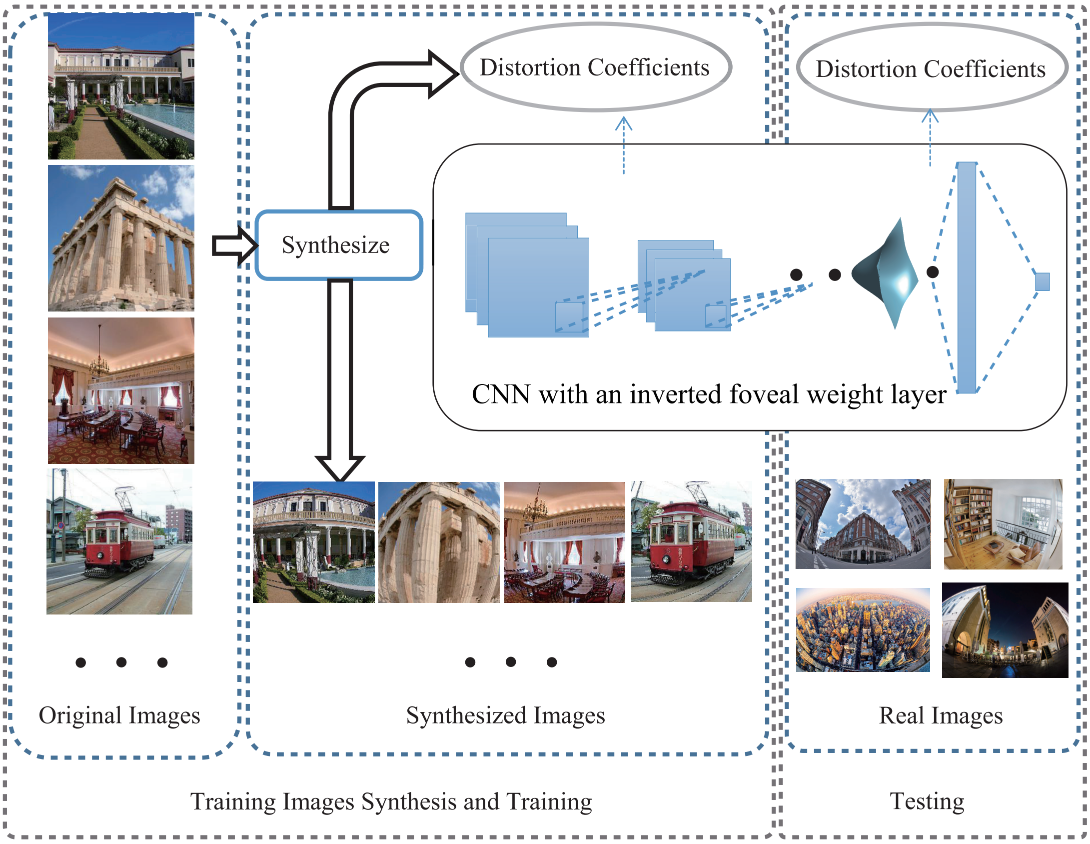

Fig. 2. System overview. We synthesize image datasets by generating radial lens distorted images simulated from real images in ImageNet [5]. The synthetic datasets are input into our CNN added by an inverted foveal layers and train it to map its synthesized distortion coefficients. The learned framework is applied to estimate the distortion coefficients of real images.

Alexnet and Resnet-18 we add an inverted foveal weight layer in the end of convolution layer, followed by fully connected layer. Feature maps output from the final convolutional layers are multiply by the inverted foveal weight layers. See Fig. 3.

There are many ways to initialized the inverted foveal weight layer. In this paper, we choose inverted gaussian kernel to initialize the inverted foveal layer. The inverted gaussian kernel is defined as:

$$f(r) = \frac{1}{\sigma\sqrt{2\pi}}(1 - e^{-\frac{r^2}{2\sigma^2}}) \tag{7}$$

where r is the distance from the image point to the image center, σ is the variance. Through this operation, we can enhance the effect of the pixel far from the image center.

D. Implementation

The distortion coefficient are continuous, so we can use Mean Square Error (MSE) to measure the network loss:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(\xi_i^{'} - \xi_i)^2 \tag{8}$$

where $\xi_i^{'}$ denotes the predict results, ξ_i denotes ground truth, n denotes batchsize.

To all experiments we fix the weight of foveal layer. We set dropout rate to 0.4 in fully connected layers. All networks are trained using Adaptive Moment Estimation (ADAM) method with weight decay 0.001 (on weight and bias). The initial learning rate is set to 0.001, and is divided by 3 every 20 epochs for Alexnet or divided by 9 every 20 epochs for Resnet-18. We train the network for 80 epochs using train datasets, and test every 1 epoch. We implement the network in pytorch architecture.

III. Results and Analysis

A. Experiment Results

In our experiment, we test four different sets of σ to initialize the inverted foveal weight layer, they are $\sigma = 0.25, \sigma = 1, \sigma = 4$, and $\sigma = +\infty$ where means we do not add any weight layers corresponding to method proposed in [15]. The train and test loss can be seen in Fig. 4, Fig. 5 and TABLE I.

From the experimental results can be seen that the train and test loss of Alexnet and Resnet-18 decrease with epoch. All of the network exist overfitting at a certain level. In Alexnet, adding inverted foveal weight layer with $\sigma = 4$ achieves the best effect. However, it works even worse when adding weight layer with $\sigma = 0.25$. In Resnet-18, adding weight layer with $\sigma = 1$ can have the best effect, and it works better when adding weight inverted foveal weight layer than not. On the whole,

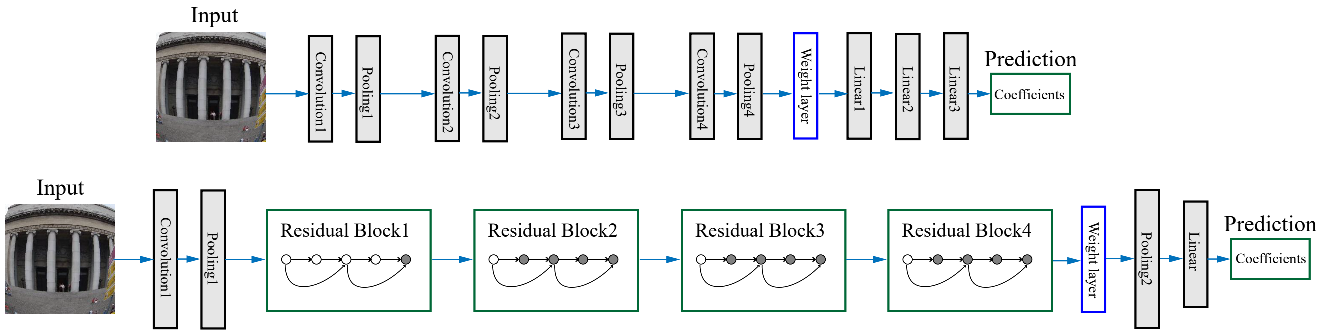

Fig. 3. Illustrate the position of a weight layer with inverted foveal models added in Alexnet (top) and Resnet-18 (bottom), respectively

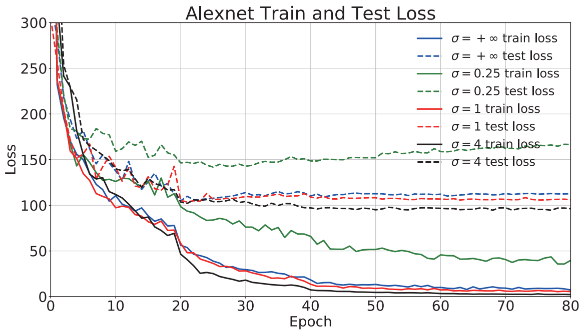

Fig. 4. Alexnet train and test loss. Note that, the method with $\sigma = +\infty$, is corresponding to the case where no layer is added, i.e., the previous method [15].

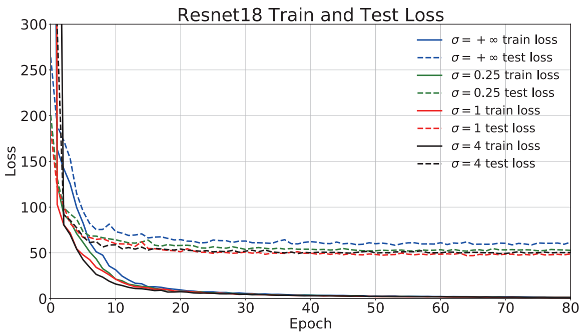

Fig. 5. Resnet-18 train and test loss. Note that, the method with $\sigma = +\infty$, is corresponding to the case where no layer is added, i.e., the previous method [15].

the results of Resnet-18 are better than Alexnet. Resnet-18 by adding inverted foveal weight layer with $\sigma = 1$ achieves the best among all experiments. By adding inverted foveal layer, the loss can be decreased by 20% than not [15]. It is obvious that in the problem of distortion correction, adding an inverted foveal weight layer can enhance the representation of the network. More results can be seen in Fig. 6 and Fig. 7.

B. Experiment Analysis

Pixels far from the image center present more distortion information, so when we add an inverted foveal weight layer,

TABLE I
LOSS CORRESPONDING TO CNNS WITH DIFFERENT σ.

Network	σ	Loss
Alexnet	0.25	160.3
	1	105.6
	4	95.7
	$+\infty$ [15]	112.2
Resnet-18	0.25	53.4
	1	**48.1**
	4	50.2
	$+\infty$ [15]	60.8

it works better than not. From the experiment results, different σ have different effects on different networks. For Resnet-18, adding $\sigma = 1$ weight layer performs better than $\sigma = 0.25$ and $\sigma = 4$, this kind of results might be related with the distribution of train distortion datasets. In Alexnet, when adding weight layer with $\sigma = 0.25$, the network works even worse than not, it might because the σ is too small and the network loss global information as well as can not highlight the difference between the image points far from the image center. It is also obvious that the Resnet-18 have stronger representation ability.

IV. CONCLUSION

Previous method shows that training CNN by massive synthetic data is an effective approach for radial distortion correction. Based upon existing image dataset, they generated millions of images with accurate radial distortion coefficients at negligible human cost. In this paper, we add a weight layer to the previous method, which is an inverted foveal model, lies in between convolution layers and fully connected layers. The weight layer only adds a small computational overhead, enabling a fast system and rapid experimentation. Our CNN based distortion coefficient estimator significantly outperforms the previous method, tested on real image dataset.

V. ACKNOWLEDGEMENT

This work was supported in part by State Key Development Program Grand No. 2016YFB1001001, NNSFC Grant No. 61322309, and NNSFC Grant No. 61273283.

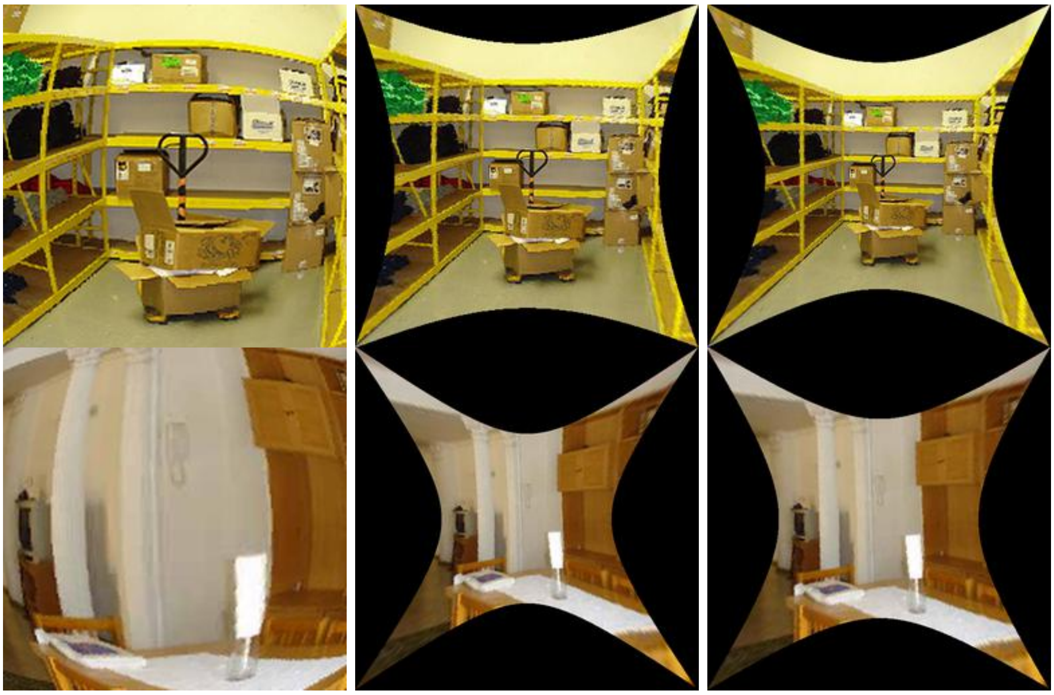

Fig. 6. Distortion correction results of some indoor scenes. Left column are the distorted images. The middle column are the corrected images using the method propose in [15]. The right column are the corrected images using our method.

REFERENCES

[1] Miguel Alemán-Flores, Luis Alvarez, Luis Gomez, and Daniel Santana-Cedrés. Line detection in images showing significant lens distortion and application to distortion correction. *Pattern Recognition Letters*, 36:261–271, 2014.

[2] Anup Basu and Sergio Licardie. Alternative models for fish-eye lenses. *Pattern recognition letters*, 16(4):433–441, 1995.

[3] Faisal Bukhari and Matthew N Dailey. Automatic radial distortion estimation from a single image. *Journal of mathematical imaging and vision*, 45(1):31–45, 2013.

[4] David Claus and Andrew W Fitzgibbon. A rational function lens distortion model for general cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 213–219, 2005.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Frederic Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine vision and applications*, 13(1):14–24, 2001.

[7] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I, 2001.

[8] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Ciaran Hughes, Patrick Denny, Martin Glavin, and Edward Jones. Equidistant fish-eye calibration and rectification by vanishing point extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2289–2296, 2010.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Yubin Kuang, Jan E Solem, Fredrik Kahl, and Kalle Astrom. Minimal solvers for relative pose with a single unknown radial distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40, 2014.

[13] Zuzana Kukelova and Tomas Pajdla. A minimal solution to radial distortion autocalibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2410–2422, 2011.

[14] Xiang Mei, Sen Yang, Jiangpeng Rong, Xianghua Ying, Shiyao Huang, and Hongbin Zha. Radial lens distortion correction using cascaded one-parameter division model. In *IEEE International Conference on Image Processing*, pages 3615–3619, 2015.

[15] Jiangpeng Rong, Shiyao Huang, Zeyu Shang, and Xianghua Ying. Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *Asian Conference on Computer Vision*, pages 35–49. Springer, 2016.

[16] Edward Rosten and Rohan Loveland. Camera distortion self-calibration using the plumb-line constraint and minimal hough entropy. *Machine Vision and Applications*, 22(1):77–85, 2011.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[18] Xianghua Ying and Zhanyi Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. *Computer Vision-ECCV 2004*, pages 442–455, 2004.

[19] Xianghua Ying, Xiang Mei, Sen Yang, Ganwen Wang, Jiangpeng Rong, and Hongbin Zha. Imposing differential constraints on radial distortion correction. In *Asian Conference on Computer Vision*, pages 384–398, 2014.

[20] Xianghua Ying, Xiang Mei, Sen Yang, Ganwen Wang, and Hongbin Zha. Radial distortion correction from a single image of a planar calibration pattern using convex optimization. In *IEEE International Conference on Image Processing*, pages 3440–3443, 2014.

[21] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 666–673, 1999.

Fig. 7. Distortion correction results of some outdoor scenes. Left column are the distorted images. The middle column are the corrected images using the method propose in [15]. The right column are the corrected images using our method.