Pattern Recognition Letters 183 (2024) 147–154

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Improving radial lens distortion correction with multi-task learning

Igor Janos [*], Wanda Benesova

Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Slovakia

ARTICLE INFO

Editor: Ricardo Torres

MSC:
41A05
41A10
65D05
65D17

Keywords:
Radial distortion
Camera calibration
Sports
Football
Multi-task learning

ABSTRACT

With computer vision and machine learning, sports image analysis can enhance viewer engagement and contribute to the overall understanding of sports events. However, beneath the apparent simplicity of sports images lies a complex challenge of radial distortion, which can impede their accurate interpretation. The need for high precision and real-time performance is paramount. We present a new regression-based method for radial distortion correction that improves the accuracy of distortion model coefficient prediction by introducing a secondary learning task that compares the distortion level of two random training samples. The secondary task requires no additional annotation, and because it is also related to radial distortion, it encourages the common feature extractor component to learn more general and robust features while preventing overfitting and improving the efficiency of training data. We have evaluated our proposed method using two public datasets and compared it against five other methods. Our method surpassed them all, both in the accuracy of the radial distortion correction and speed as well. You can find the source code and trained models at https://vgg.fiit.stuba.sk/improving-radial.

1. Introduction

Sports image analysis is one of the domains that has significantly benefited from advances in computer vision. With the growing adoption of machine learning methods, new products and tools have emerged that make sports image analysis accurate and accessible for professional athletes, amateur and junior athletes, and fans worldwide. Sports images are crucial visual records, aiding post-game or live analysis and decision-making. However, they are often affected by radial distortion that hinders their accurate interpretation and utilization. In a study, presented by Vieira et al. [1], the authors report, that the mean uncertainty of a player tracking system operating on images affected by radial distortion can reach up to 20 cm, while observing a futsal court with the typical dimensions of 33×18 m.

In the past, radial distortion correction has relied on geometric techniques or camera calibration processes. Such calibration was typically performed in laboratory conditions using special calibration patterns. Once appropriately calibrated, the camera parameters had to remain constant.

In the sports domain, we can encounter a diverse set of cameras ranging from long-distance cameras that follow the play slowly, wide-angle action cameras mounted on top of referees' helmets, to cable-suspended cameras capable of both horizontal and vertical movement over the playfield. To properly utilize images captured by these cameras, we must be able to accurately and in real-time remove radial distortion, which changes rapidly on a frame-by-frame basis.

In this paper, we propose a deep-learning regression-based method to rectify images containing radial lens distortion that operates on single independent images (Fig. 1). Our method has been trained on the football domain dataset and has been fine-tuned specifically for images containing minor radial distortion, which is more difficult to rectify accurately and is typical for television footage of football games.

Compared to other deep-learning radial distortion correction methods, we introduce in this paper, the following contributions:

- *Secondary learning task* that learns useful features on random combinations of sample pairs (Section 3.3).
- *Penalty term* that encourages better accuracy on low distortion images (Section 3.4).

2. Related work

2.1. Camera model

In our work, we assume the pinhole camera model as described in [2], which projects straight lines in the real world into straight lines in the observed images. 3D points $p_{3d} = (X, Y, Z)$ are projected into a plane located at $Z = 1$, and the 2D planar coordinates of the projected points $p = (x, y) = (X/Z, Y/Z)$ are called the *normalized image coordinates*. When you scale the normalized image coordinates by the

[*] Corresponding author.
E-mail addresses: igor.janos@stuba.sk (I. Janos), vanda_benesova@stuba.sk (W. Benesova).

https://doi.org/10.1016/j.patrec.2024.05.008
Received 27 October 2023; Received in revised form 8 April 2024; Accepted 11 May 2024
Available online 17 May 2024

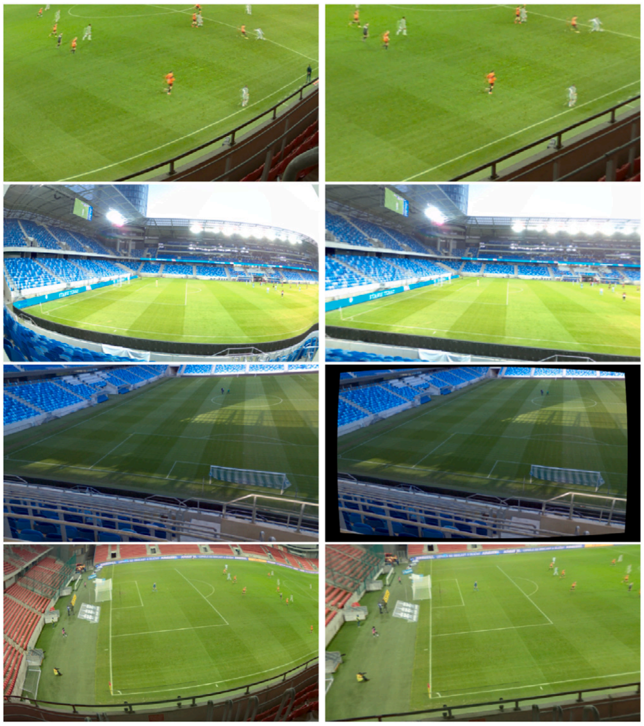

Fig. 1. Distorted images (left column) and images corrected by our method using the estimated distortion coefficients (right column).

focal length f, you can obtain the *pixel coordinates* $p_i = (u, v) = (fx, fy)$, which are relative to the image sensor center.

2.2. Radial distortion models

Images captured by real-world cameras often contain a certain amount of radial distortion, which is proportional to the distance from the image center. There are two distortion models dominant in the literature. The first is the polynomial model proposed by Duane [3], expressed by the equation:

$$\boldsymbol{x} = (1 + k_1 \|\boldsymbol{p}\|^2 + k_2 \|\boldsymbol{p}\|^4 + k_3 \|\boldsymbol{p}\|^6 + \cdots)\boldsymbol{p} \quad (1)$$

where $k_1, k_2, k_3, \ldots$ are the radial distortion parameters (or coefficients). The other is the division model proposed by Fitzgibbon [4], expressed by the equation:

$$\boldsymbol{p} = \frac{1}{(1 + \lambda_1 \|\boldsymbol{x}\|^2 + \lambda_2 \|\boldsymbol{x}\|^4 + \lambda_3 \|\boldsymbol{x}\|^6 + \cdots)}\boldsymbol{x} \quad (2)$$

where $\lambda_1, \lambda_2, \lambda_3, \ldots$ are the coefficients of the model.

The level of precision you can achieve by using any of these models is determined by the number of coefficients you wish to use. For many applications, using just one coefficient might be enough. The single-parameter division model is generally easier to work with and has the nice property of mapping straight lines into circular arcs.

2.3. Distortion correction methods

Classical methods. Bukhari and Dailey [5] have proposed to fit circles into detected line segments. In [6], the authors have proposed the Improved Hough Transform, which can detect distorted lines, and in [7], the authors have proposed a method for radial distortion correction using the Improved Hough Transform and a single-parameter division model. In [8], an extension to the modified Hough transform was proposed, which can iteratively optimize two-parameter radial distortion models, both polynomial and division.

Regression-based learning methods. Rong et al. [9] have used a classification approach to estimate the distortion parameter with a finite precision as one of 401 possible discrete values. They have synthesized training data from a subset of the ImageNet dataset, proposed by Russakovsky et al. [10], that satisfied the condition of containing strong line segments detectable by the Hough transform. Lopez et al. [11] have successfully trained the convolutional neural network to jointly estimate the camera orientation (tilt, roll) and intrinsic parameters (focal length and radial distortion) from single images. Their method used synthetic data generated from panoramic images of the SUN360 dataset, presented by Xiao et al. [12], which contain man-made objects of everyday life. The method estimated two distortion parameters of the polynomial model, expressing the k_2 parameter as a function of k_1. Liao et al. [13] have extended the framework proposed by Rong et al. [9] by adding a new representation for the lens distortion called the *Ordinal Distortion*. Instead of estimating the distortion coefficients directly, the authors try to estimate the distortion rate for the image features with gradually increasing distance from the image center, the shape of the distortion function sampled at given radii.

Reconstruction-based learning methods. We can find several methods that approach distortion correction as an image-to-image translation problem. Liao et al. [14] have used adversarial training to train a GAN, which models mapping between the distorted and rectified images. In the work of Li et al. [15] and Chao et al. [16], and Zhu et al. [17], the authors have tried to improve the interpretability of the predictions by incorporating the correction flow. Li et al. [15] have fitted a single-parameter division model to the predicted correction flow. In [16], the authors have learned a geometric prior to improve the consistency of the correction flow. In [17], the authors have explored a transformer-based UNet architecture for the correction flow prediction. In [18], the authors propose a complementary network that contains parallel branches for correction flow prediction and distortion correction. At each level of the distortion decoder, the proper estimate of the correction flow is used to produce undistorted features. In [19], the authors propose a model-aware pre-training, where a transformer-inspired encoder learns to predict the distortion type and distortion parameters. After the pre-training is finished, the rectification decoder is fine-tuned to predict the correction flow.

2.4. Metrics

When evaluating the accuracy of radial distortion correction, it feels natural to use L_1, or L_2 distance between the ground truth and approximated coefficients. However, different groups of coefficients may yield similar levels of the distortion effect, which makes distance metrics suboptimal for accuracy evaluations. Also, it is not possible to compare the performance of methods built using different distortion models. Liao et al. [13] have proposed a new mean distortion level deviation metric (MDLD),

$$MDLD = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} |\hat{d}(i,j) - d(i,j)| \quad (3)$$

where W and H are the width and height of the image, and $\hat{d}(i,j)$ is the distortion yielded by the approximated coefficients of the given pixel, and $d(i,j)$ is the ground truth distortion. This metric is independent of the chosen radial distortion model.

Additionally, we will also be using the structural similarity index measure (SSIM), introduced by Wang et al. [20], and peak signal-noise ratio (PSNR) metrics, described in [21], to compare the undistorted images using ground truth and estimated coefficients. For two single-channel grey-level 8-bit images, where f is the reference image and g is the test image, both of size $H \times W$ pixels, the PSNR is defined by:

$$PSNR(f,g) = 10 \log_{10} \frac{255^2}{MSE(f,g)} \quad (4)$$

where

$$MSE(f,g) = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}(f_{ij} - g_{ij})^2 \tag{5}$$

For color images, the PSNR value can be calculated as the average of the PSNR of all individual color channels.

The SSIM is defined by:

$$SSIM(f,g) = l(f,g)\cdot c(f,g)\cdot s(f,g) \tag{6}$$

where

$$\begin{aligned} l(f,g) &= \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \\ c(f,g) &= \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \\ s(f,g) &= \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \end{aligned} \tag{7}$$

The luminance comparison function $l(f,g)$ measures how close the mean image luminances μ_f, μ_g are. The contrast comparison function $c(f,g)$ takes into account the standard deviation of the two images σ_f, σ_g. The third structure comparison function $s(f,g)$ measures the correlation coefficient between the two images f, g. The positive constants C_1, C_2, C_3 are to prevent null denominator.

3. Proposed method

Dataset. We will be using the *Football360* dataset, introduced by Jánoš and Benesova [22], which was created specifically for the task of radial distortion correction. The dataset consists of 268 equirectangular panoramas in 16384×8192 pixels resolution that were created using the Panono panoramic camera. The panoramic camera has a spherical shape with 12 cm in diameter with 36 fixed-focus cameras spread evenly over the spherical surface. Each of the 36 cameras has a 2-megapixel sensor. Images from all of the 36 sensors are stitched together using a cloud stitching service provided by the Panono company. The Panono company has not disclosed the inner details of their stitching algorithm, but because the panoramic camera contains only fixed-focus cameras, and the radial lens distortion correction has been a part of the panorama stitching process since the work of Sawhney and Kumar [23], we *assume*, that the Panono stitching algorithm handles radial distortion as well. In the *Football360* dataset, there are also four convenient export sets of cropped views available, which were created from the equirectangular panoramas by randomly sampling pan, tilt, roll, and field-of-view. The sets A, B, C are meant for training, and contain $\{30,000; 100,000; 300,000\}$ images. The set V is meant for validation and contains 10,000 images. The authors have first rendered the images in 1920×1080 and then downscaled them into 448×448 pixels resolution. All exported sets (A, B, C, V) from the *Football360* dataset were distorted using a 2-parameter polynomial model. The distribution of the distortion coefficients follows the discovery of Lopez et al. [11], which states that for many real cameras the k_1, k_2 coefficients are closely correlated, and adds gaussian noise $\epsilon \sim N(0, 0.02^2)$ to the k_2 coefficient to allow for more variety in the observed distortion effects.

Baseline. As the baseline (Fig. 2), we have selected the regression-based method described in [22], which is based on the work of Lopez et al. [11]. The baseline method consists of a convolutional backbone feature extractor and a single regressor head estimating the single k_1 distortion coefficient. Lopez et al. [11] have decided to use the DenseNet-161 architecture, proposed by Huang et al. [24], as the backbone feature extractor. In [22], the authors have evaluated multiple backbones - ResNet-152 (proposed by He et al. [25]), DenseNet-161 (proposed by Huang et al. [24]), and EfficientNet-B5 (proposed by Tan and Le [26]). Reconstruction-based methods, which directly produce undistorted images, typically operate in lower resolution, and as such, they are not practical for our desired use case. However, in [15], the coefficient of a single-parameter division model is recovered by fitting a model to the predicted low-resolution correction flow and this allows us to analytically compute the distortion even for high-resolution images. Lopez et al. [11] have discovered that for many real-world cameras the values of k_1 and k_2 coefficients are closely correlated, and a good estimate of k_2 can be obtained by:

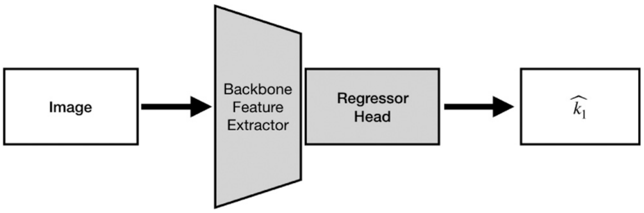

Fig. 2. The baseline method from [22] estimates the single k_1 distortion parameter.

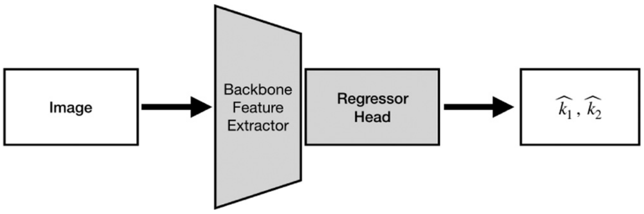

Fig. 3. The main learning task of our proposed method is the direct estimation of the k_1, k_2 distortion coefficients.

$$k_2 = 0.019k_1 + 0.805k_1^2 \tag{8}$$

In our method, we extend the original design of the baseline method and estimate both coefficients independently. We also explore techniques that provide additional gains in accuracy and training data efficiency. In our experiments, we also explore a range of smaller backbone feature extractors - EfficientNet-B2, EfficientNet-B0, and EfficientNet-LiteB0.

All evaluated feature extractors were pre-trained on the *ImageNet* classification task. The native input resolution for all of them is 224×224 pixels with 3 color channels. We only use layers from *conv1* until *conv5_x* for the ResNet extractors. For the DenseNet extractors, we only use layers from the first convolutional layer until the final Dense Block (4). And for the EfficientNet extractors, we use stages from 1 to 8. We have omitted the final global average pooling and classification head from all architectures. The extracted feature tensor has 7×7 spatial resolution. The regressor head of our network has an optional global average pooling, a single hidden layer followed by batch normalization and ReLU activation, and a final output layer with one or two units, depending on the number of coefficients we wish to estimate. We have discussed the effect of the optional global average pooling in more detail in Section 4.1.1.

3.1. Regression of 2 coefficients

We propose a method that directly estimates both coefficients of the 2-parameter polynomial radial distortion model k_1, k_2 (Fig. 3). Compared to the baseline method of Jánoš and Benesova [22], having the freedom to independently estimate both distortion coefficients should prove beneficial.

3.2. MDLD loss

When working with a 2-parameter distortion model, we can observe two important facts:

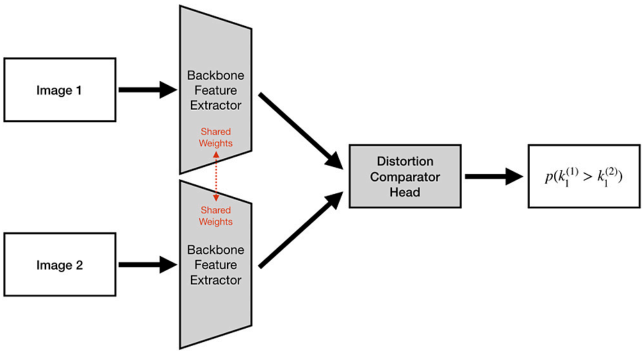
Image 1
Image 2
Backbone Feature Extractor
Shared Weights
Shared Weights
Backbone Feature Extractor
Distortion Comparator Head
$p(k_1^{(1)} > k_1^{(2)})$

Fig. 4. The secondary learning task of our proposed method predicts the probability that the k_1 coefficient of the first image is larger than the k_1 coefficient of the second image.

- different combinations of k_1, k_2 can lead to similar distortion effects
- very often, the distortion effect is dominated by the k_1 coefficient

Because of this, we propose to directly use the MDLD (Eq. (3)) which combines the effect of both distortion model parameters into a single metric as the training objective function. Let $m(i,j,k_1,k_2) = m(\mathbf{x},k_1,k_2) = 1 + k_1\|\mathbf{x}-\mathbf{c}\|^2 + k_2\|\mathbf{x}-\mathbf{c}\|^4$ be a function of the distortion model yielding the rate of how much a 2D point $\mathbf{x}$ (i,j are the individual components of the 2D vector) is to be distorted, where $\mathbf{c}$ is the distortion center. Note that we are operating in normalized screen coordinates.

The MDLD loss for the pair of ground truth values (k_1,k_2) and the given estimates $(\widehat{k_1},\widehat{k_2})$ can be calculated as

$$\mathcal{L}_{MDLD}((\widehat{k_1},\widehat{k_2}),(k_1,k_2)) = \frac{1}{WH}\sum_{i}^{H}\sum_{j}^{W}|m(i,j,\widehat{k_1},\widehat{k_2}) - m(i,j,k_1,k_2)| \tag{9}$$

To mitigate the problem, where the k_1 dominates the distortion, we evaluate the contribution of each of the components separately as

$$\mathcal{L}_{k_1}((\widehat{k_1},\widehat{k_2}),(k_1,k_2)) = \mathcal{L}_{MDLD}((\widehat{k_1},k_2),(k_1,k_2)) \tag{10}$$

$$\mathcal{L}_{k_2}((\widehat{k_1},\widehat{k_2}),(k_1,k_2)) = \mathcal{L}_{MDLD}((k_1,\widehat{k_2}),(k_1,k_2)) \tag{11}$$

3.3. Secondary learning task

We have observed that all feature extractors in our experiments tend to overfit the training data. To address this issue, we propose a simple regularization technique (Fig. 4), which we call the *Distortion Level Comparison* (DLC). For any distorted image, the value of k_1 coefficient determines what kind of radial distortion can we observe with the barrel distortion in the negative range of k_1 and the pincushion distortion in the positive range of k_1. Using the feature extractor sharing weights with the main task, we extract feature vectors from all samples in our mini-batch and use them to produce randomly selected feature vector pairs. We then train a comparator head, which is a separate network component, to predict which feature vector of the given pair exhibits stronger distortion effect, determined by the ground truth k_1 values. By doing so, we present the model with diverse combinations of samples and encourage the common feature extractor to learn more generalized and robust features.

For a mini-batch $\mathcal{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ containing m samples, we apply the feature extractor function $F(x)$ and compute a set of features $\mathcal{F} = \{f^{(1)}, f^{(2)}, \dots, f^{(m)}\}$, where $f^{(i)} = F(x^{(i)})$. The set $\mathcal{K}_1 = \{k_1^{(1)}, k_1^{(2)}, \dots, k_1^{(m)}\}$ contains ground truth values of k_1 for the given mini-batch samples $\mathcal{X}$. The set $\mathcal{I} = \{1,2,\dots,m\}$ is a set of m ordered indices. And the set $\mathcal{J}$ is a random permutation of $\mathcal{I}$. The comparator head, represented by the function $\mathcal{C}(f^{(i)}, f^{(j)})$, is trained for all pairs (i,j), where i is ith element of $\mathcal{I}$ and j is the ith element of $\mathcal{J}$ to predict the probability $p(k_1^{(i)} > k_1^{(j)})$ using the binary cross-entropy loss. The comparator part of the loss function $\mathcal{L}_C(\mathcal{F},\mathcal{K}_1)$ is thus defined as

$$\mathcal{L}_C(\mathcal{F},\mathcal{K}_1) = E_{(i,j)}\left[\mathcal{L}_{BCE}(\mathcal{C}(f^{(i)}, f^{(j)}), p(k_1^{(i)} > k_1^{(j)}))\right] \tag{12}$$

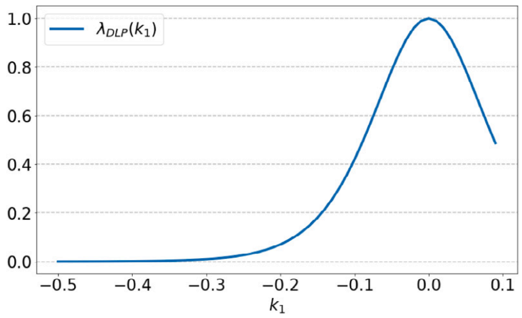

Fig. 5. Plot of $\lambda_{DLP}(k_1, a = 10)$.

3.4. Distortion level penalty term

In [22], the authors have pointed out that minor distortion seems more difficult to estimate than significant distortion. To put more emphasis on accurately estimating minor distortion, we propose a penalty function $\lambda_{DLP}(k_1)$ that is dependent on k_1. A good candidate for such a function might be the first derivative of tanh,

$$\frac{d\tanh(x)}{dx} = \frac{1}{\cosh^2(x)} \tag{13}$$

To better match our desired k_1 range, we propose to use a parametrized version (Fig. 5),

$$\lambda_{DLP}(k_1, a) = \frac{1}{\cosh^2(ax)} \tag{14}$$

where $a = 10$.

For the set $\widehat{\mathcal{K}_1} = \{\widehat{k}_1^{(1)}, \widehat{k}_1^{(2)}, \dots, \widehat{k}_1^{(m)}\}$ containing $\widehat{k_1}$ predictions we can compute the distortion level penalty part of the loss function as square error between the ground truth k_1 and the predicted $\widehat{k_1}$ multiplied by the penalty function $\lambda_{DLP}(k_1)$:

$$\mathcal{L}_{DLP}(\widehat{\mathcal{K}_1},\mathcal{K}_1) = E\left[\mathcal{L}_{MSE}(\widehat{k_1},k_1)\cdot\lambda_{DLP}(k_1)\right] \tag{15}$$

3.5. Horizontal flip

We can notice that the radial distortion is symmetrical, and equal distortion coefficients should be associated with a pair of horizontally mirrored images. We can use this fact to our advantage and calculate the estimated coefficients as a mean of coefficients obtained from the original and mirrored input image during the inference time. This way, we can get a slight increase in accuracy at the cost of higher computational complexity.

3.6. The final training objective

The final training objective can be obtained by adding all the loss components together as,

$$\mathcal{L} = \mathcal{L}_{k_1} + \mathcal{L}_{k_2} + \lambda_1\mathcal{L}_C + \lambda_2\mathcal{L}_{DLP} \tag{16}$$

where λ_1, λ_2 are weight coefficients balancing the contribution of individual loss components. We recommend training with $\lambda_1 = 0.2$, and $\lambda_2 = 30$.

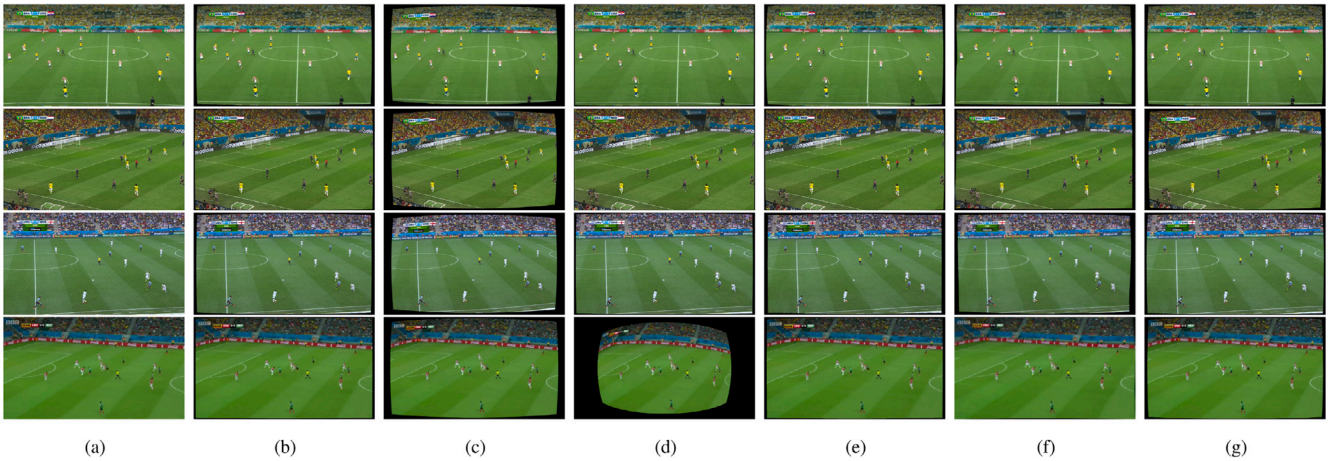

Fig. 6. Qualitative evaluation on the WorldCup 2014 dataset. Best viewed when zoomed in. Input images (a), corrected images by our method (b), corrected images by Jánoš and Benesova [22] (c), corrected images by Li et al. [15] (d), corrected images by Santana-Cedrés et al. [8] (e), corrected images by Yang et al. [18] (f), and corrected images by Wang et al. [19] (g)

Table 1
The final results of our method evaluated on the Football360 dataset.

Method	SSIM ↑	PSNR ↑	MDLD ↓
Baseline (ResNet-152 w. GA) Jánoš and Benesova [22]	0.852	23.075 dB	0.0172
Li et al. [15]	0.417	10.919 dB	1.431
Santana-Cedrés et al. [8]	0.714	21.647 dB	0.1132
Yang et al. [18]	0.561	17.362 dB	–
Wang et al. [19]	0.560	15.109 dB	0.305
Our method	**0.933**	**30.098 dB**	**0.0088**

Table 2
Performance evaluation on 186 images from the *WorldCup 2014* dataset. Each image was processed as an individual batch of size 1.

Method	Total time ↓	Per frame ↓
Baseline (ResNet-152 w. GA) Jánoš and Benesova [22]	14.161 s	76.1 ms
Li et al. [15]	40.71 s	218.8 ms
Santana-Cedrés et al. [8]	1856 s	9978.5 ms
Yang et al. [18]	10.93 s	58.75 ms
Wang et al. [19]	9.534 s	51.2 ms
Our method	**9.231 s**	**49.6 ms**

4. Evaluation

We have performed a quantitative evaluation of the performance of our method and three other methods on the validation set of the *Football360* dataset (Table 1). The method of Li et al. [15] struggled the most. The predicted correction flow was often inaccurate, and the subsequent single-parameter division model fitting ended up with a completely inaccurate estimate of the distortion parameter. Moreover, the single-parameter division model has proven inadequate in capturing the subtle distortion typical for football images. The method of Santana-Cedrés et al. [8] has performed reasonably well thanks to its two-parameter polynomial model, but struggled on images, where the playing field lines were not distinctively visible. It is also worth mentioning that it was by far the slowest method with a processing time of nearly 3 s per image. The method of Wang et al. [19] was slightly worse than [8]. The model-free reconstruction-based method of Yang et al. [18] uses the correction flow only internally, and produces undistorted images in an end-to-end fashion. This makes it impossible to evaluate using the MDLD metric. To obtain fair SSIM and PSNR results, we have retrained the method with rescaled training images in 640 × 360 resolution, which was also the resolution we used for SSIM and PSNR calculation for the rest of the methods. This gave the [18] method an advantage of much higher input resolution. However, if it were to be used for high-resolution images, it would need to be retrained again. The method of Jánoš and Benesova [22] was consistently more accurate on difficult images than [8] but could not capture the fine radial distortion defined by the k_2 parameter. Our proposed method outperformed all other evaluated methods by a significant margin.

In most cases, the proposed method worked well (Figs. 1, 6). By inspecting the samples on which our method returned the largest errors, we have found that the *Football360* validation set contains images that do not contain a reasonable view of the football field (Fig. 7). These images are either over-exposed, or are the results of pointing the imaginary camera directly to the tribune seats during dataset generation. It is worth mentioning that even the fail cases of our method are still quite good when compared to other methods. The worst recorded MDLD value on *Football360* dataset was 0.154 (first image of Fig. 7), which is still comparable to the overall performance of Santana-Cedrés et al. [8].

We have also performed a qualitative comparison (Fig. 6) of all the mentioned methods on the *WorldCup 2014* dataset, introduced by Homayounfar et al. [27]. The images in *WorldCup 2014* are from the main broadcast camera and are easier than *Football360*, meaning that they are in 1280 × 720 resolution, and most of them contain clearly visible playing field lines. None of the *WorldCup 2014* images were used during the training of neither of the evaluated methods.

We have measured the time it took for each of the evaluated methods to process all of the 186 images (Table 2) contained in the *WorldCup 2014* dataset using a PC equipped with a single AMD Ryzen 5900 CPU and a single GeForce 3090 RTX GPU. The method of Santana-Cedrés et al. [8] was implemented in C++ and operates in the full 1280 × 720 resolution. The rest of the evaluated methods were implemented in Python using the *PyTorch* library. We have used the *OpenCV* library to decode and rescale the images. In the method of Li et al. [15], the undistorted images were produced by the proposed iterative resampling procedure. For the rest of the methods, the undistorted images were remapped using the image coordinate maps computed according to the estimated distortion model coefficients or correction flow.

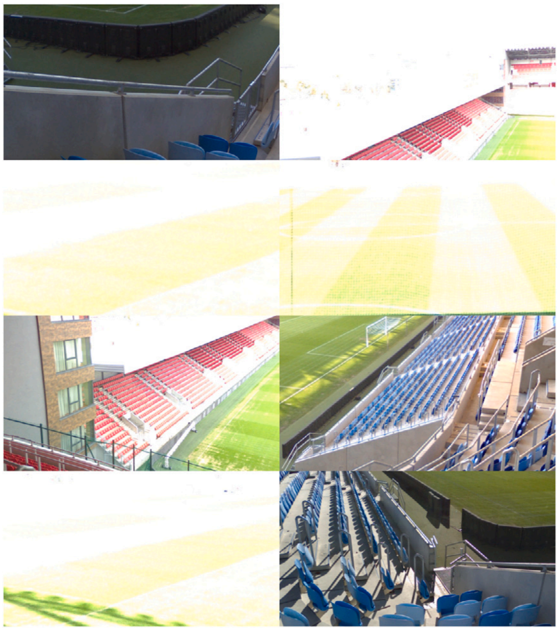

Fig. 7. Images from the *Football360* dataset, on which our method fails.

Table 3
Contribution of individual components of our method. The components of our method are (a) EfficientNet-Lite0 backbone with a regressor head with two output units, (b) Distortion Level Comparison (DLC), (c) Distortion Level Penalty term (DLP), (d) Horizontal flipping during inference time (HF).

Method	Coeff.	MDLD ↓
Baseline (ResNet-152 w. GA)	1	0.0172
Lite0	2	0.0113
Lite0 + DLC	2	0.0095
Lite0 + DLC + DLP	2	0.0093
Lite0 + DLC + DLP + HF	2	**0.0088**

4.1. Ablation study

In this section, we explore our experiments and the contribution of each of the components of our method (Table 3).

4.1.1. Estimating 2 coefficients

We first compare the effect of estimating two independent distortion coefficients. We must emphasize one important difference between our further experiments and the reference method of Jánoš and Benesova [22]: global average pooling at the end of the feature extractor backbone. Larger models estimating a single coefficient worked better with global average pooling followed by a larger hidden layer. Smaller backbone models worked better without the global average pooling and with a smaller hidden layer in the regressor head. In this experiment (Fig. 8), we turn off the global average pooling, change the number of output units to 2, and keep all other aspects exactly as in [22]. We choose EfficientNet-B5 as our backbone feature extractor. We can observe that even though the achieved MDLD is much lower, the training is rather unstable, and the model has overfitted on the training data. From now on, we do not use global average pooling anywhere.

Next, we try reducing the size of the feature extractor backbone. We experiment with EfficientNet-B2, EfficientNet-B0, and EfficientNet-Lite0 (Fig. 9). It seems that smaller models work better, and all tested backbones (Table 4) smaller than EfficientNet-B5 converge on roughly the same level of MDLD, which is significantly lower than achieved by the single-parameter baseline method. Note that the smallest tested model had only 16 units in the hidden layer.

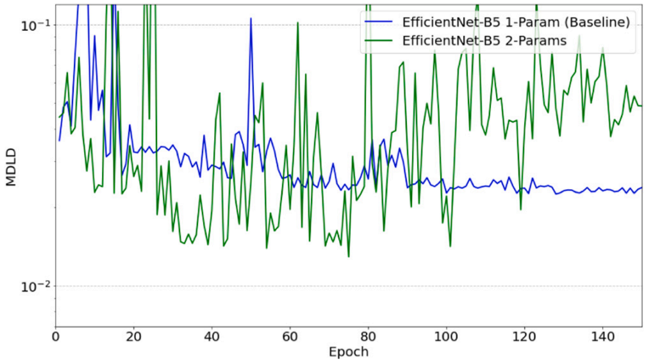

Fig. 8. Training with two distortion parameters is much more volatile than with only a single parameter.

Table 4
Properties of the single-parameter baseline model and tested 2-parameter models.

Backbone	Coeff.	Units	MDLD ↓
EfficientNet-B5 (w/o GA)	1	256	0.0225
EfficientNet-B5 (w. GA)	1	256	0.0173
EfficientNet-B5	2	256	0.0129
EfficientNet-B2	2	256	0.0109
EfficientNet-B0	2	256	**0.0109**
EfficientNet-Lite0	2	16	0.0113

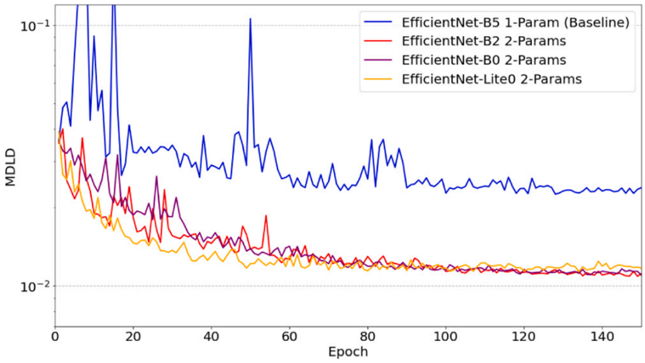

Fig. 9. Smaller feature extractor backbones perform better.

4.1.2. Effect of distortion level comparison

The comparator head network component, described in Section 3.3, consists of a fully connected hidden layer containing just 16 units, followed by batch normalization, swish activation, and a single output unit. Thanks to the random selection of image pairs, it is much harder for the feature extractor to overfit on the main task. In the areas of stronger distortion, our model can predict the correct distortion coefficients more easily. By introducing the additional learning task, we can produce difficult sample pairs, even in the areas that were considered easy from the main task perspective, and we can force the feature extractor to learn more general and robust distortion features (Fig. 10).

4.1.3. Effect of distortion level penalty

We have found out that we can obtain small gains by properly tuning the λ_2 weight coefficient that controls the contribution of the distortion level penalty function. Large λ_2 values make the model sacrifice precision in stronger distortion images. The $\lambda_2 = 30$ value seems to be optimal and results in small improvements both in the low-distortion part of the k_1 range (Fig. 11, Table 5), as well as in the overall MDLD.

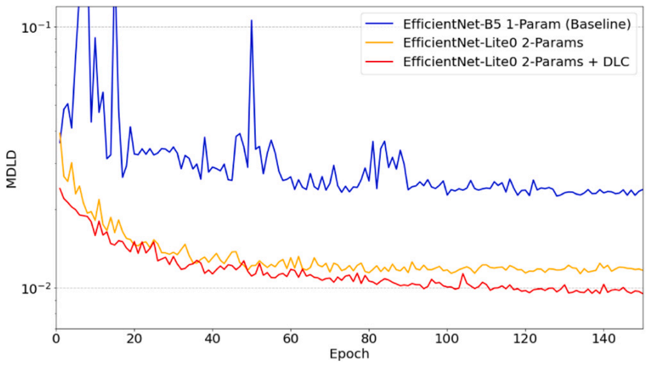

Fig. 10. Training of the smallest EfficientNet-Lite0 model with the *Distortion Level Comparison* regularization.

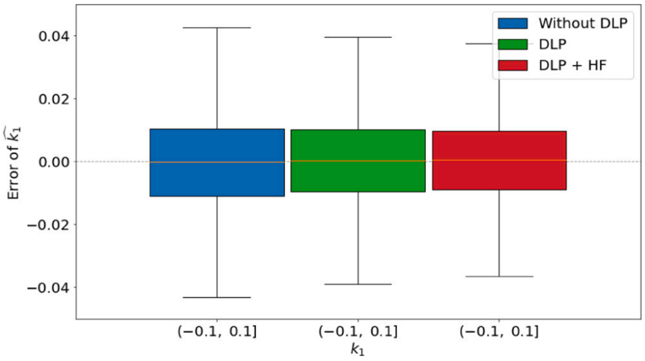

Fig. 11. Quantiles of the distributions of $\hat{k_1}$ error for low-distortion interval of k_1.

Table 5
Quantiles of the distributions of $\hat{k_1}$ error for low-distortion interval of k_1.

Setup	Q_1 (25%)	Q_3 (75%)
Lite0 + DLC	−0.01107	0.01043
Lite0 + DLC + DLP	−0.00966	0.01003
Lite0 + DLC + DLP + HF	**−0.00897**	**0.00960**

5. Conclusion

In this paper, we have explored new ways of improving radial distortion correction methods. We have focused specifically on the sports domain, namely football. We have successfully trained a model to predict two independent coefficients of the two-parameter polynomial radial distortion model, which has proven to be more accurate than single-parameter models, or models where the second coefficient is modeled as a function of the first coefficient. We showed that in the football domain, smaller feature extractor architectures exhibit less training volatility. By including the secondary learning task, we have managed to reduce the overfitting problem further, increase the data efficiency during the training, and improve the overall accuracy of our model. Because of the simplicity of our method, and the use of a smaller feature extractor, our method is significantly faster than other evaluated methods and can be considered real-time. In future work, we plan to explore how we can apply our proposed radial distortion correction method to the camera calibration problem.

CRediT authorship contribution statement

Igor Janos: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation. **Wanda Benesova:** Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] L.H. Vieira, E.A. Pagnoca, F. Milioni, R.A. Barbieri, R.P. Menezes, L. Alvarez, L.G. Déniz, D. Santana-Cedrés, P.R. Santiago, Tracking futsal players with a wide-angle lens camera: accuracy analysis of the radial distortion correction based on an improved hough transform algorithm, Comput. Methods Biomech. Biomed. Eng.: Imaging Vis. 5 (3) (2017) 221–231.

[2] R. Klette, A. Koschan, K. Schluns, Three-Dimensional Data from Images, Springer-Verlag Singapore Pte. Ltd., Singapore, 1998.

[3] C.B. Duane, Close-range camera calibration, Photogramm. Eng. 37 (8) (1971) 855–866.

[4] A.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. CVPR 2001, Vol. 1, IEEE, 2001, p. I.

[5] F. Bukhari, M.N. Dailey, Automatic radial distortion estimation from a single image, J. Math. Imaging Vis. 45 (1) (2013) 31–45.

[6] M. Alemán-Flores, L. Alvarez, L. Gomez, D. Santana-Cedrés, Line detection in images showing significant lens distortion and application to distortion correction, Pattern Recognit. Lett. 36 (2014) 261–271.

[7] M. Alemán-Flores, L. Alvarez, L. Gomez, D. Santana-Cedrés, Automatic lens distortion correction using one-parameter division models, Image Process. Line 4 (2014) 327–343.

[8] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, L. Alvarez, An iterative optimization algorithm for lens distortion correction using two-parameter models, Image Process. Line 6 (2016) 326–364.

[9] J. Rong, S. Huang, Z. Shang, X. Ying, Radial lens distortion correction using convolutional neural networks trained with synthesized images, in: Asian Conference on Computer Vision, Springer, 2016, pp. 35–49.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[11] M. Lopez, R. Mari, P. Gargallo, Y. Kuang, J. Gonzalez-Jimenez, G. Haro, Deep single image camera calibration with radial distortion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11817–11825.

[12] J. Xiao, K.A. Ehinger, A. Oliva, A. Torralba, Recognizing scene viewpoint using panoramic place representation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 2695–2702.

[13] K. Liao, C. Lin, Y. Zhao, A deep ordinal distortion estimation approach for distortion rectification, IEEE Trans. Image Process. 30 (2021) 3362–3375.

[14] K. Liao, C. Lin, Y. Zhao, M. Gabbouj, DR-GAN: Automatic radial distortion rectification using conditional GAN in real-time, IEEE Trans. Circuits Syst. Video Technol. 30 (3) (2019) 725–733.

[15] X. Li, B. Zhang, P.V. Sander, J. Liao, Blind geometric distortion correction on images through deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4855–4864.

[16] C.-H. Chao, P.-L. Hsu, H.-Y. Lee, Y.-C.F. Wang, Self-supervised deep learning for fisheye image rectification, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2020, pp. 2248–2252.

[17] F. Zhu, S. Zhao, P. Wang, H. Wang, H. Yan, S. Liu, Semi-supervised wide-angle portraits correction by multi-scale transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19689–19698.

[18] S. Yang, C. Lin, K. Liao, C. Zhang, Y. Zhao, Progressively complementary network for fisheye image rectification using appearance flow, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6348–6357.

[19] W. Wang, H. Feng, W. Zhou, Z. Liao, H. Li, Model-aware pre-training for radial distortion rectification, IEEE Trans. Image Process. (2023).

[20] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[21] A. Hore, D. Ziou, Image quality metrics: PSNR vs. SSIM, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 2366–2369.

[22] I. Jánoš, W. Benesova, Football360: Introducing a new dataset for camera calibration in sports domain., in: VISIGRAPP (4: VISAPP), 2023, pp. 301–308.

[23] H.S. Sawhney, R. Kumar, True multi-image alignment and its application to mosaicing and lens distortion correction, IEEE Trans. Pattern Anal. Mach. Intell. 21 (3) (1999) 235–243.

[24] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[26] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[27] N. Homayounfar, S. Fidler, R. Urtasun, Sports field localization via deep structured models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5212–5220.