applied optics

Direct distortion prediction method for AR-HUD dynamic distortion correction

Fangzheng Yu,[1] Nan Xu,[1] Shiqi Chen,[1] Huajun Feng,[1] Zhihai Xu,[1] Qi Li,[1] Tingting Jiang,[2] and Yueting Chen[1,*]

[1]State Key Laboratory of Modern Optical Instrumentation, College of Optical Science and Engineering, Zhejiang University, Hangzhou 310027, China
[2]Research Center for Intelligent Sensing, Zhejiang Laboratory, Hangzhou 311100, China
*chenyt@zju.edu.cn

Received 10 April 2023; revised 12 June 2023; accepted 28 June 2023; posted 28 June 2023; published 14 July 2023

Dynamic distortion is one of the most critical factors affecting the experience of automotive augmented reality head-up displays (AR-HUDs). A wide range of views and the extensive display area result in extraordinarily complex distortions. Existing methods based on the neural network first obtain distorted images and then get the predistorted data for training mostly. This paper proposes a distortion prediction framework based on the neural network. It directly trains the network with the distorted data, realizing dynamic adaptation for AR-HUD distortion correction and avoiding errors in coordinate interpolation. Additionally, we predict the distortion offsets instead of the distortion coordinates and present a field of view (FOV)-weighted loss function based on the spatial-variance characteristic to further improve the prediction accuracy of distortion. Experiments show that our methods improve the prediction accuracy of AR-HUD dynamic distortion without increasing the network complexity or data processing overhead.

https://doi.org/10.1364/AO.492602

1. INTRODUCTION

The on-vehicle augmented reality head-up display (AR-HUD) system uses the optical system to project the input image onto the car's front windshield, forming a virtual image at a certain distance in front of the vehicle, and the driver obtains enhanced information by observing the projected image [1,2]. However, this process undergoes a complicated optical conversion of the image and the non-standard shape of the front windshield. The displayed image will have spatial position offsets and severe distortion, making the virtual image different from the original image, as shown in Fig. 1(a). It is necessary to calibrate and predistort the projected image so that the observed image is consistent with the original one, as shown in Fig. 1(b). At present, the existing technologies used in automobiles select the best observation position as a fixed observation point and only calibrate the projected image of the fixed viewpoint [3–5]. However, since the position of the driver's or passenger's viewpoint changes dynamically, the projected image corrected by a fixed position may appear distorted, which cannot fully meet the experience needs of the viewer.

There are some studies on AR-HUD distortion correction for dynamic viewpoint.

A. Traditional Dynamic Distortion Correction

Wientapper *et al*. [6] proposed dynamic distortion correction for vehicle AR-HUD system. They estimated the projective transformation from the world to the calibration camera and compensated the optical distortion by introducing a fifth-degree polynomial model. However, it is limited to a small display area, and the calibration process involves complex components. Ueno and Komuro [7] raised another calibration method based on multi-view. They created a conversion map from a calibrated camera image to the virtual image for each viewpoint. Then they employed linear regression to create a lookup table (LUT) for different viewpoints. Deng *et al*. [8] implemented the AR-HUD calibration using mixed reality glasses. They sampled hundreds of viewpoints and applied nonlinear regression to estimation coefficients. Like [7], Gao *et al*. [9–12] also used LUT to record the predistortion mapping table of calibrated viewpoints, obtaining the map of any viewpoint by LUT interpolation. However, these methods require a large amount of data to form a LUT or estimate regression coefficients, which has specific requirements for system storage, especially with the improvement of image resolution and precision requirements. At the same time, linear interpolation may cause image hopping [13], negatively impacting the user experience.

B. Dynamic Distortion Correction Based on Deep Learning

The method obtains a dynamic predistortion model through deep learning, and the model can predict the predistortion

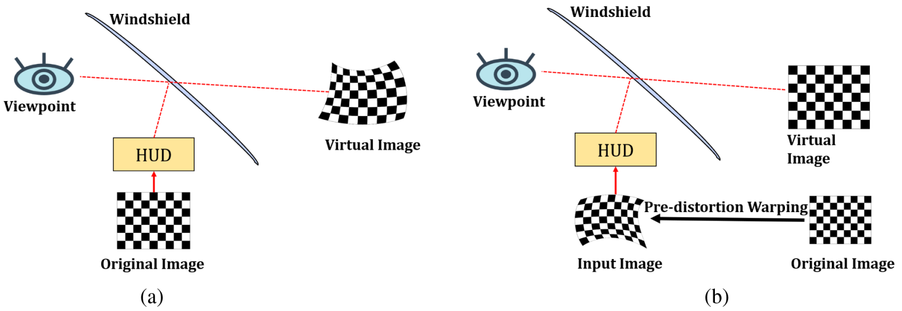


Fig. 1. AR-HUD produces distortion when the original image is projected onto the windshield. To ensure that the virtual image is not distorted, the original image should be predistorted. (a) projects the original image directly, causing image distortion, while (b) projects the original image with a predistortion warping according to the calibration information to correct the distortion in (a).

correction parameters of any viewpoint to achieve image correction. Compared with traditional methods, neural network methods have superior generalization ability and processing efficiency. It helps to eliminate the problems caused by interpolation and has broad application. Li *et al*. [13] proposed multilayer feedforward neural network model and spatial continuous mapping (MFNN-SCM) for dynamic distortion correction of AR-HUD. This method constructs a seven-layer fully connected network. The network inputs are the viewpoint coordinates and the ideal point coordinates, and the output is the predistorted point coordinates corresponding to the ideal points. The ideal-distorted point pairs on different viewpoints are obtained through actual shooting and image processing, and then linear interpolation is used to obtain the ideal-predistortion point pairs as the network dataset. Root mean square error (RMSE) is the loss function for training and evaluating the model. After training, the model outputs the predistorted coordinates of the corresponding points of any viewpoint and generates the predistorted image through forward mapping. This method realizes the distortion correction for any viewpoint and obtains good imaging results in both the marked viewpoints and the non-marked viewpoints.

C. Other Works

Head-mounted displays (HMDs) share many similarities with HUDs as they also use a translucent reflector as a combination to create virtual images in front of the eyes. There are several methods [14–16] available for calibrating and correcting vertical image distortion. As HMDs concern more gaze direction and the eye position is fixed, it does not provide a useful reference for the HUD.

Li's approach generates a predistortion model by predicting the predistortion coordinates and using forward mapping to rectify the image. However, in the process of data acquisition, what we capture through the camera is distorted images, so interpolation processing is required to obtain predistorted coordinates, which will introduce additional time consumption and coordinate errors. In this paper, we propose to directly use the distortion data to train a distortion model and correct the image by predicting the distortion coordinates and backward mapping. Using our pipeline saves time and the cost of data processing and avoids coordinate errors caused by interpolation.

In addition, the network design and training of previous methods do not fully reflect the property of distortion. Distortion in optics refers to the offset between the real image and the ideal image and generally increases with the field of view (FOV). Therefore, we first predict the distortion offsets and then add ideal coordinates to obtain the final distortion coordinates. In this way, the accuracy of distortion prediction is improved without increasing the network complexity. Besides, we propose a loss function design method based on the weight of the FOV, which can improve the model's ability to predict distortion. Our work is prone to realize real-time dynamic distortion correction in AR-HUD and provide consumers with a better display experience.

In summary, our contribution includes the following:

1. A direct distortion prediction method based on neural network for AR-HUD dynamic distortion correction. The distorted data are used to train the network model, and the predistorted images are then obtained through backward mapping. This manner significantly decreases the computational overhead and avoids the interpolation errors.
2. We propose to predict the distortion offsets instead of the distortion coordinates, which improves prediction accuracy without increasing the network complexity.
3. We propose a loss function based on the weight of FOV. Different penalties are applied to FOVs to adapt to the spatial-variance of dynamic distortion.

Experiments show that our methods can reduce time consumption and improve the dynamic distortion correction effect of AR-HUD. The proposed methods are general and can be applied in the method of AR-HUD dynamic distortion correction based on the neural network.

2. METHOD

A. Direct Distortion Model

As shown in Fig. 2(a), the predistortion prediction model of distortion correction includes HUD dataset construction, predistortion network training and prediction, and predistorted image generation. First, multiple distorted images under set viewpoints are shot to obtain ideal-distortion point pairs. In order to train the predistortion model, it is necessary to convert

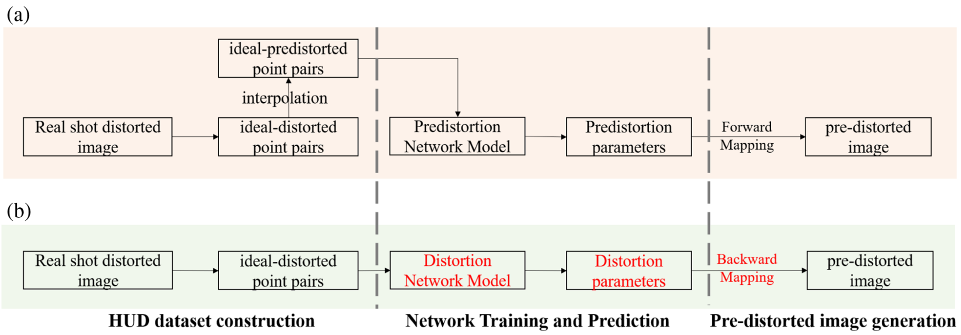

(a)
ideal-predistorted point pairs
interpolation
Real shot distorted image
ideal-distorted point pairs
Predistortion Network Model
Predistortion parameters
Forward Mapping
pre-distorted image
(b)
Real shot distorted image
ideal-distorted point pairs
Distortion Network Model
Distortion parameters
Backward Mapping
pre-distorted image
HUD dataset construction
Network Training and Prediction
Pre-distorted image generation

Fig. 2. Predistortion prediction model for AR-HUD dynamic distortion correction and our proposed direct distortion prediction model. The predistortion prediction model obtains the ideal-predistortion point pairs to train the model and predicts the predistortion parameters. Our method directly trains the distortion model, predicts the distortion parameters, and then performs backward mapping to generate a predistorted image.

the ideal-distortion point pairs into ideal-predistortion ones through interpolation. After training, the predistortion coordinates of each point predicted by the model are used to generate predistorted images via forward mapping. This method needs to interpolate distortion point pairs, which increases additional time and calculation costs and introduces interpolation errors. In addition, forward mapping may generate hole pixels [17].

To solve these two problems, we propose a direct distortion prediction model. As shown in Fig. 2(b), we use the ideal-distorted point pairs to directly train the network model. Unlike the predistortion model, this model predicts the distortion coordinates of the input points. When getting each point distortion coordinates under the input viewpoint, we predistort the image through backward mapping. It should be noted that this method does not need to adjust the neural network model but only uses the distorted data as the training dataset. Figure 3 shows the overall process. The specific data will be shown in Section 3.

B. Predict the Distortion Offsets

Distortion in optics refers to the deviation from rectilinear projection in which straight lines of an image appear to be curved unnaturally. For a two-dimensional image point, it can be expressed as Eqs. (1) and (2),

$$\delta\mu = \mu_d - \mu_i, \tag{1}$$

$$\delta\nu = \nu_d - \nu_i, \tag{2}$$

where (μ_i, ν_i) are the coordinates of the ideal image point, (μ_d, ν_d) are the coordinates of the distorted image point, and $(\delta\mu, \delta\nu)$ are the distortion offsets. Generally, the distortion coordinates are very close to the ideal ones because the distortion offset is a small value. Figures 4(a) and 4(b) show the ideal image point coordinates (μ_i, ν_i), Figs. 4(c) and 4(d) show the distorted image point coordinates (μ_d, ν_d), and Figs. 4(e) and 4(f) show the distortion offsets $(\delta\mu, \delta\nu)$. It can be seen that the real image point surface is similar to the ideal one, and its distortion cannot be clearly expressed. Obviously, the distortion offset surface can better represent the distortion of different image points. For this reason, we optimize the existing network architecture to predict the distortion offsets instead of the distortion coordinates.

The network architecture in the general AR-HUD predistortion prediction model is shown in Fig. 5(a). The inputs of the network are the viewpoint and ideal image point coordinates, and the output is the predistorted image point coordinates. This architecture predicts the predistorted coordinates from the ideal coordinates.

Here we propose to predict the distortion offsets through the network. Based on Eqs. (1) and (2), we add the input ideal coordinates to the network output, which means that the network predicts the distortion offsets. Compared with the general method, our method is more in line with the meaning of distortion. Experiments prove that this architecture can improve the prediction accuracy of distortion without complicated data processing or adjusting the network model. The specific results are shown in Section 3.

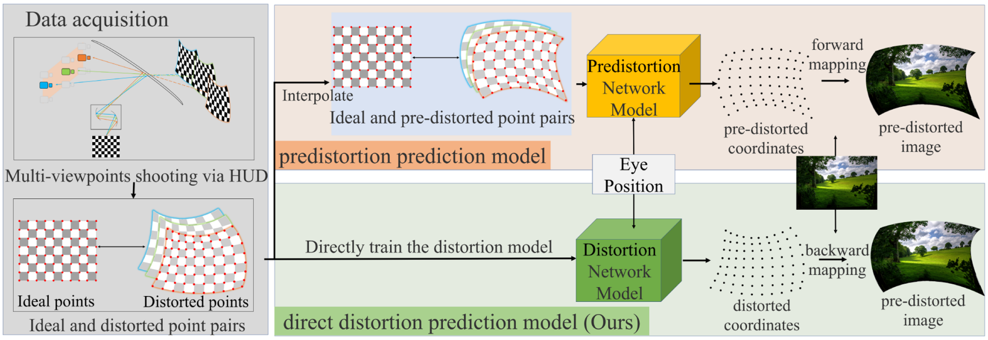

Data acquisition
Multi-viewpoints shooting via HUD
Ideal points
Distorted points
Ideal and distorted point pairs
Interpolate
Ideal and pre-distorted point pairs
Predistortion Network Model
pre-distorted coordinates
forward mapping
pre-distorted image
predistortion prediction model
Eye Position
Directly train the distortion model
Distortion Network Model
distorted coordinates
backward mapping
pre-distorted image
direct distortion prediction model (Ours)

Fig. 3. Schematic diagram of the predistortion prediction model and our proposed direct distortion prediction model.

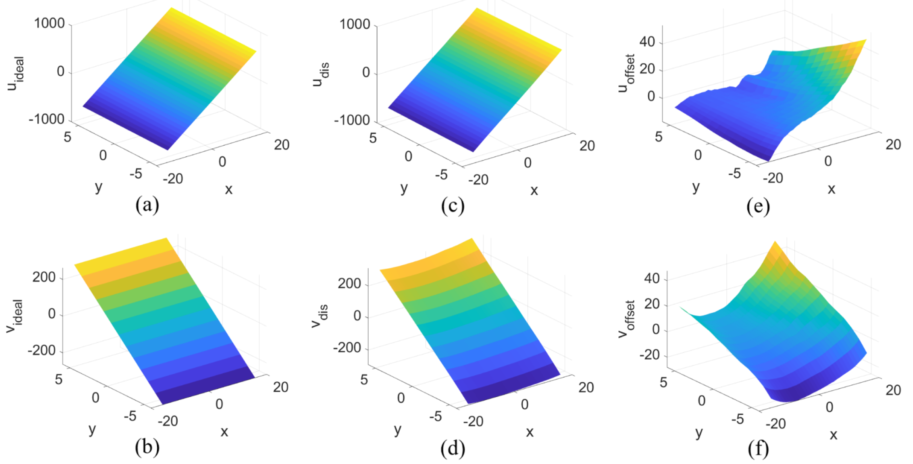
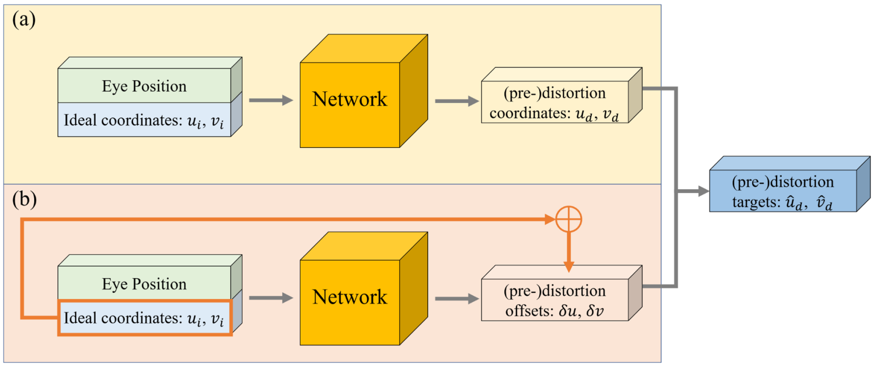


Fig. 4. Ideal image point coordinate, real image point coordinate, and distortion offset of the real image point under the same viewpoint. x and y are the position subscripts. (a) Coordinate value of the ideal image point μ_i; (b) coordinate value of the ideal image point ν_i; (c) coordinate value of the real image point ν_d; (d) coordinate value of the real image point μ_d; (e) distortion offset of the real image point $\delta\mu$; (f) distortion offset of the real image point $\delta\nu$.



(a)
Eye Position
Ideal coordinates: u_i, v_i
Network
(pre-)distortion coordinates: u_d, v_d
(pre-)distortion targets: $\hat{u}_d$, $\hat{v}_d$
(b)
Eye Position
Ideal coordinates: u_i, v_i
Network
(pre-)distortion offsets: δu, δv

Fig. 5. This figure shows the network structures of predicting the distortion coordinates and distortion offsets. Compared with distortion prediction, the offset prediction network adds the input ideal coordinates to the network output as the final output, which means that the network predicts the distortion offsets. This improvement provides our method with better performance but the same complexity as previous network.

C. FOV-Weighted Loss Function

The distortion offset of each point is related to the FOV. Figures 6(a) and 6(b) show the ideal grid points and the distorted grid points, and Fig. 6(c) shows the percentage of distortion of different FOVs of these two images. The calculation formula is shown in Eq. (3),

$$\text{distortion} = \sqrt{\frac{\delta\mu^2 + \delta\nu^2}{\mu^2_{\text{halfField}} + \nu^2_{\text{halfField}}}}, \qquad (3)$$

where $\mu_{\text{halfField}}$, $\nu_{\text{halfField}}$ are the maximum half-FOV in both μ, ν directions. Here, all the distortions are divided by the maximum half-FOV because in the AR-HUD projection we are more concerned about each absolute offset of the distorted pixel. We can see that, with the increase of the FOV, the percentage of distortion gradually increases. So, the pixels in the central FOV need not be corrected or only need a minor correction, while larger adjustments are required to reduce distortion for the large FOV. This should be treated differently during network training.

In Li's method, the RMSE is used, and the network loss function is

$$\mathcal{L}_{\text{MFNN}-\text{FCM}} = \frac{1}{m \cdot n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\begin{pmatrix}\hat{u}_{i,j}\\ \hat{v}_{i,j}\end{pmatrix} - \begin{pmatrix}u_{i,j}\\ v_{i,j}\end{pmatrix}\right\|^2, \qquad (4)$$

where μ, ν are ideal coordinate values of grid points, $\hat{\mu}$, $\hat{\nu}$ are actual coordinate values, m is the number of the viewpoints, and n is the number of grid points in a single image. This function adopts the same loss calculation for each point. To reflect attention to different FOV points, we propose a loss function design method based on FOV weights. The function is

$$\mathcal{L}' = \mathcal{L} \cdot W_{\text{field}}, \qquad (5)$$

$$W_{\text{field}_{i,j}} = \alpha\begin{pmatrix}u_{ij}^2/\max(u^2)\\ v_{ij}^2/\max\left(v^2\right)\end{pmatrix} + \beta, \qquad (6)$$

where $\mathcal{L}$ is the initial loss, $\mathcal{L}'$ is the FOV-weighted loss, W_{field} is the weight coefficient related to the FOV, and α and β are constant items that can be set. This loss function introduces the

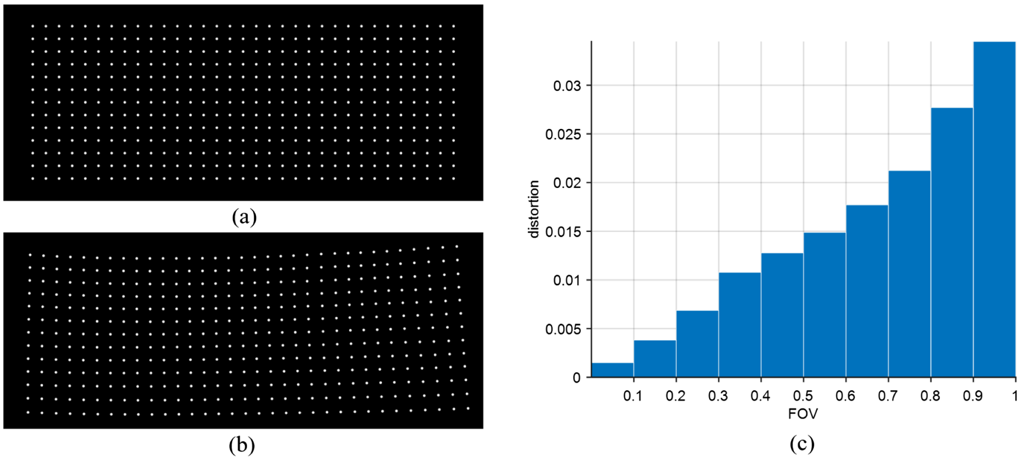


Fig. 6. (a) and (b) are the ideal grid points and the distorted grid points; (c) is the percentage of distortion of different FOVs of these two images.

FOV information into the training process and takes the penalty for the distortion of different FOVs. The experimental results show that the network trained by our FOV-weighted loss has higher prediction accuracy. See Section 3 for specific data.

3. EXPERIMENTAL RESULTS

We will present our experimental results in this section. First, we use ZEMAX to build an AR-HUD system to obtain distortion data for training and testing. Then we construct our network model based on the methods we proposed. Finally, we compare the accuracy of the distortion prediction and time consumption with Li's method.

A. Dataset

In the AR-HUD system environment of ZEMAX simulation, we used the image simulation function to obtain distorted images of multiple viewpoints within the eye box for model training and testing. We select an area in the $X - Y$ plane that covers typical driver's eye positions. Since the opening angles of the virtual image to any viewpoint are relatively small and the focal length is designed several meters in ARHUDs, the distortion variation is insensitive to the depth shift of viewpoints [9]. As shown in Fig. 7, the size of the eye box is 40 × 100 mm, and the total number of set viewpoints is 5 × 11. The data of 3 × 3 viewpoints are used for training, and the remaining viewpoints are used for testing. The input image is the grid points image shown in Fig. 6(a), the number of dots is 13 × 33, and the image resolution is 1920 × 1080. Through image processing steps such as point recognition and corner sorting, we can obtain the HUD dynamic distortion dataset,

$$\left\{E_x^k, E_y^k, u_i^k, v_i^k, u_d^k, v_d^k\right\}, (k = 1, 2, 3, \cdots, n), \quad \textbf{(7)}$$

where (E_x^k, E_y^k), (u_i^k, v_i^k), (u_d^k, v_d^k) are the viewpoint coordinate, ideal point coordinate, and distortion coordinate of the kth point.

B. Model Construction

We construct the direct distortion prediction model with distortion offsets prediction and FOV-weighted loss. To better compare with Li's method, we use the same MLP as the initial network. First, as shown in Fig. 2, the initial predistortion model is changed to a direct distortion prediction model; second, by adding the input ideal coordinates to the network output, the network structure is improved to predict distortion offsets. Finally, as shown in Eq. (6), the loss function is changed to the FOV-weighted loss. We used the same optimizer and hyper-parameters in these models to ensure the fairness of the experimental results. The results after each step of the methods are tested, and the predistortion correction effect of the model on all viewpoints is calculated. Specifically, we re-input the predistorted image into the HUD as a simulated image and calculate the distortion percentage as the correction result for comparison.

C. Experimental Results

The experimental network environment is the Windows system, the GPU is RTX 2060, and the experimental platform is PyTorch. We use Adam with an initial learning rate of 1×10^{-3}. The models are trained with a batch size of 429 (the number of grid points in a single image) for 4500 iterations.

Since distortion offset is a small quantity, we generally use the percentage of distortion to characterize the distortion of an image. For the 13 × 33 points under each viewpoint, we can obtain the maximum and average distortion of the viewpoint by Eq. (3). We also calculated the maximum error and the RMSE of the actual error (at 1080 × 1920 resolution) in μ, ν directions to more comprehensively test the predistortion correction result of the model.

Table 1 shows the results of Li's and our method. We start experiments with the initial MLP network (Li's method). In the first step, we add the direct distortion prediction process to the model (Model 1); then, we change the network structure to predict the distortion offsets (Model 2). Finally, we add the FOV weight to the original MSE-Loss (Model 3). Here max $\_d\%$ indicates the maximum distortion (expressed as a percentage),

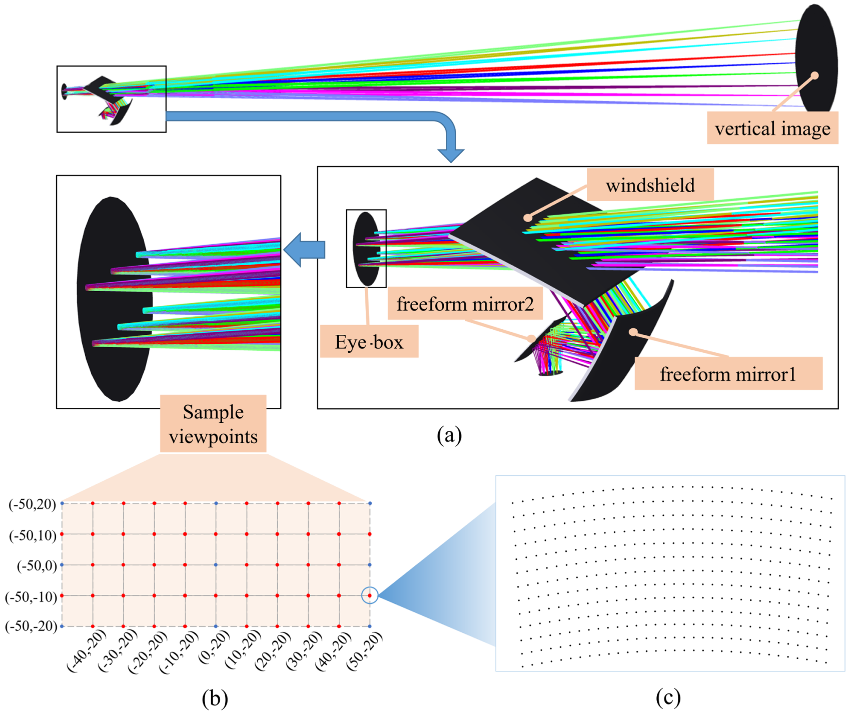


Fig. 7. ZEMAX simulation system and samples of the data of 5 × 11 viewpoints. The blue viewpoint is the training data, and the red viewpoint is the test data.

Table 1. Experiment Results of Li's and Our Method

Model	Max Max_d%	Mean Max_d%	Max Mean_d%	Mean Mean_d%	ME_μ	ME_ν	$RMSE_\mu$	$RMSE_\nu$
Li's method	4.79	2.61	2.35	1.06	37.42	14.20	7.24	2.78
Model 1	4.77	2.51	2.03	0.96	37.12	13.17	6.73	2.23
Model 2	1.55	**0.99**	0.70	0.32	12.00	9.27	1.93	1.24
Model 3 (ours)	**1.53**	1.00	**0.69**	**0.31**	**11.74**	**8.87**	**1.89**	**1.22**

and mean_d% indicates the average distortion at a certain viewpoint. In addition, the maximum and RMSE error of μ and ν are also shown in Table 1.

Table 1 shows that our methods can improve the predistortion correction accuracy of the AR-HUD. Each method will improve the accuracy to a certain extent, and the optimal correction accuracy can be achieved when all three methods are used. In addition, we can find that the biggest improvement in accuracy is predicting the distortion offsets, which can better reflect the dynamic distortion of HUD. From the data in Table 1, our method works better both for distortion percentage and actual pixel error.

To verify that the direct distortion prediction model can decrease the time consumption, according to the standard in Fig. 2, we divided the overall process into dataset construction, network training, and image predistortion and compared the time consumption of two models.

The data in Table 2 are the average value of multiple experimental results. As shown in Table 2, since the predistortion prediction model needs to interpolate the distorted image to obtain a predistorted image and our model directly uses the distorted image, Li's method takes 8.70 s to construct the dataset while ours only takes 4.87 s. In the network training phase, the time used by the two is basically the same because the distortion data or predistortion data does not affect the network itself. In the image predistortion phase, our method uses a faster backward mapping and only needs 22 ms (for a piece of image at 1920 × 1080), compared to Li's 26 ms. Therefore, Table 2 verifies that the direct distortion prediction model can significantly reduce the time consumption of the overall process and has advantages in the distortion correction of AR-HUD, saving the time consumption and avoiding the coordinate errors in interpolation. Predistortion image generation reaches speed in 30 fps at 1080p in our experimental environment. Theoretically with 4.5 MFLOPs of our model, calculating the distortion parameters of 6 × 6 feature points reaches 16.2 ms at 10 GFLOPS. It will help deploy the network model to on-vehicle systems.

Table 2. Time Consumption of Predistortion Prediction Model and Direct Distortion Prediction Model

Model	Dataset Construction	Network Training	Image Predistortion
Predistortion prediction model (Li's)	8.70 s	162.72 s	26 ms
Direct distortion prediction model (ours)	**4.87 s**	162.83 s	**22 ms**

4. CONCLUSION

This paper proposes a direct distortion prediction method based on neural network for AR-HUD dynamic distortion correction. We directly train the neural network with the distorted image data. The method does not require predistortion interpolation, saving time consumption and avoiding coordinate errors in interpolation. Besides, based on the physical priors and characteristics of distortion, we propose to predict the distortion offsets of each viewpoint in the eye box. Moreover, the proposed field-of-view weighted loss improves the prediction ability of the model. Experiments show that, without increasing network complexity and data processing capacity, the proposed method benefits the prediction accuracy of the neural-based distortion correction overall. It is helpful to realize real-time correction of dynamic distortion. In the future, we plan to further prune and optimize these methods, and deploy them to specific equipment.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data will be made available on request.

REFERENCES

1. H. S. Park, M. W. Park, K. H. Won, K.-H. Kim, and S. K. Jung, "In-vehicle AR-HUD system to provide driving-safety information," ETRI J. **35**, 1038–1047 (2013).
2. T. Poitschke, M. Ablassmeier, G. Rigoll, S. Bardins, S. Kohlbecher, and E. Schneider, "Contact-analog information representation in an automotive head-up display," *Symposium on Eye Tracking Research & Applications*, New York, New York, USA, March 26, 2008, (Association for Computing Machinery) pp. 119–122.
3. W. Wu, F. Blaicher, J. Yang, T. Seder, and D. Cui, "A prototype of landmark-based car navigation using a full-windshield head-up display system," *Workshop on Ambient Media Computing*, New York, New York, USA, October 23, 2009, (Association for Computing Machinery) pp. 21–28.
4. W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, J. Wieghardt, H. Hörtner, and C. Lindinger, "A new visualization concept for navigation systems," *User-Centered Interaction Paradigms for Universal Access in the Information Society: 8th ERCIM Workshop on User Interfaces for All, Revised Selected Papers 8*, Vienna, Austria, June 28, 2004 (Springer), pp. 440–451.
5. A. Wang, T. Qiu, and L. Shao, "A simple method of radial distortion correction with centre of distortion estimation," J. Math. Imaging Vis. **35**, 165–172 (2009).
6. F. Wientapper, H. Wuest, P. Rojtberg, and D. Fellner, "A camera-based calibration for automotive augmented reality head-up-displays," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2013), pp. 189–197.
7. K. Ueno and T. Komuro, "[Poster] Overlaying navigation signs on a road surface using a head-up display," in *IEEE International Symposium on Mixed and Augmented Reality* (2015), pp. 168–169.
8. N. Deng, Y. Zhou, J. Ye, and X. Yang, "A calibration method for on-vehicle ar-hud system using mixed reality glasses," in *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018), pp. 541–542.
9. X. Gao, J. Werner, M. Necker, and W. Stork, "A calibration method for automotive augmented reality head-up displays based on a consumer-grade mono-camera," in *IEEE International Conference on Image Processing (ICIP)* (2019), pp. 4355–4359.
10. X. Gao, J. Werner, M. Necker, and W. Stork, "A calibration method for automotive augmented reality head-up displays using a chessboard and warping maps," Proc. SPIE **11433**, 114332W (2020).
11. X. Gao, M. Necker, and W. Stork, "A low-complexity yet accurate calibration method for automotive augmented reality head-up displays," Proc. SPIE **11605**, 116050B (2021).
12. X. Gao, K. Wu, M. Necker, W. Stork, A. Jadid, and G. Klinker, "A target-free calibration method for automotive augmented reality head-up displays," Proc. SPIE **11605**, 116051V (2021).
13. K. Li, L. Bai, Y. Li, and Z. Zhou, "Distortion correction algorithm of ar-hud virtual image based on neural network model of spatial continuous mapping," in *IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (2020), pp. 178–183.
14. M. Tuceryan, Y. Genc, and N. Navab, "Single-point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality," in *IEEE and ACM International Symposium on Augmented Reality (ISAR)* (2002), Vol. **11**, pp. 259–276.
15. F. Kellner, B. Bolte, G. Bruder, U. Rautenberg, F. Steinicke, M. Lappe, and R. Koch, "Geometric calibration of head-mounted displays and its effects on distance estimation," IEEE Trans. Vis. Comput. Graph. **18**, 589–596 (2012).
16. Y. Hiroi, K. Someya, and Y. Itoh, "Neural distortion fields for spatial calibration of wide field-of-view near-eye displays," Opt. Express **30**, 40628–40644 (2022).
17. D. Rueckert and P. Aljabar, "Non-rigid Registration Using Free-form Deformations," in *Handbook of Biomedical Imaging* (Springer, 2015), pp. 277–294.