ISSN: 2664-9640

Image Enhancement based Stereo Camera Calibration Method

Lixin Zhao[1, a], Fanliang Bu[1, *], Xu Wang[1], Zhou Yao[2]

[1] Department of Information Network Security, People's Public Security University University, Beijing, China

[2] School of Information Network Security, People's Public Security University, Beijing, China

[a]929225320@qq.com [*]bufanliang@sina.com

Abstract

Camera calibration is a crucial step in stereo vision 3D reconstruction, and the quality of the calibration directly affects the outcome of the 3D reconstruction. Traditional camera calibration techniques suffer from inaccuracies in corner detection, which impacts the results of camera calibration. Therefore, this paper designs a stereo camera calibration method based on deep learning corner detection. UNet is used as the benchmark network for corner detection, and innovations are made on top of this model. To address the issue of corner detection being easily interfered with by the chessboard pattern background, an image reconstruction enhancement module is introduced to reduce background features and highlight the chessboard corner features in the recognition image. To tackle the problems of false positives and missed detections in corner detection, a multi-level attention module is introduced to enhance the model's ability to distinguish between different feature points, thereby reducing errors and missed detections of chessboard corners. A multi-scale spatial attention module is introduced to increase the model's sensitivity to chessboard corner features. Real images are input into the trained model to generate a heatmap containing the corner positions. A Gaussian surface fitting algorithm is used to extract sub-pixel level corner positions from the heatmap. Finally, the Zhang Zhengyou calibration method is used for stereo camera calibration. Experimental results show that the improved modules proposed in this paper enhance the accuracy of corner detection, and the stereo camera calibration based on deep learning corner detection demonstrates certain advantages.

Keywords

Camera Calibration; Deep Learning; Corner Detection; Sub-pixel; Gaussian Surface Fitting; 3D Reconstruction.

1. Introduction

Camera calibration is a key step in stereo vision 3D reconstruction [1], whose purpose is to obtain the intrinsic and extrinsic parameters of the camera and the distortion coefficients. The quality of the calibration directly affects the outcome of the 3D reconstruction [2]. Existing calibration methods mainly fall into two categories: traditional calibration methods and self-calibration methods [3]. Both essentially estimate the camera's specific parameters through the relationship between real-world feature points and pixel coordinates. Self-calibration methods [4] do not require a specific scene and are highly practical, but due to the numerous interference factors in real-world scenes, their calibration accuracy and stability are not as good as traditional calibration methods.

Traditional calibration methods use specific calibration objects to obtain the camera's parameters, with these objects typically having distinct features such as corners, lines, or circles that can be accurately identified and located in the images captured by the camera. During

ISSN: 2664-9640

calibration, the correspondence between the measured positions of the feature points of the calibration object in the image and their actual positions in the physical world is established. This correspondence can be represented by establishing a spatial coordinate system, in which each feature point of the calibration object has its corresponding three-dimensional spatial coordinates. The Direct Linear Transformation (DLT) method proposed by Abdel-Aziz [5] and Karara is a camera calibration method based on the principles of photogrammetric geometry. This method's main characteristic is to solve for the fundamental matrix by constructing homogeneous coordinates of two-dimensional points and three-dimensional points, with the fundamental matrix describing the projection relationship between two planes. According to the camera's internal parameters, the essential matrix E is derived to obtain the camera's position relationship in the world coordinate system, followed by decomposing the camera's external parameters to obtain the rotation matrix and translation vector. The DLT method is simple to calculate and does not require iteration, but it does not consider the nonlinear distortion effects of the camera lens. In practical applications, the nonlinear distortion of the camera lens is inevitable, so more complex methods or iterative processes are usually needed to correct these distortions and improve calibration accuracy. Tsai [6] proposed a two-step method with radial constraint in 1986, which extracts corners from calibration images. Using the image coordinates of these corners and the actual coordinates of the calibration board, the camera's parameters can be obtained.

Around 2000, the Zhang Zhengyou calibration method [7] was proposed, which establishes equations using the properties of the homography matrix and rotation matrix by capturing multiple chessboard images from different angles, and solves these equations to obtain the camera's accurate parameters. Compared to other traditional methods, it is easier to operate and has higher accuracy than self-calibration methods. However, traditional calibration methods still have many shortcomings, such as inaccurate corner detection and susceptibility to noise. In recent years, with the development of deep learning and artificial intelligence, camera calibration technology has also made significant progress [8].

In 2003, Lv Zhaohui and others proposed three neural network structures for stereo vision camera calibration: direct mapping, lens distortion correction, and spatial position compensation. However, these methods are all based on shallow neural networks and cannot meet the calibration requirements in complex environments with significant distortion. Using deep learning technology, camera parameters can be automatically learned from images [9], [10], [11], thereby improving calibration accuracy. Although these methods are more robust, there are still limitations in terms of pixel-level accuracy. Recently, Chen et al. [12] proposed a chessboard corner detection method that refines the peaks of the neural network response map to obtain sub-pixel level corner positions. This method has achieved good results, but its detection network is trained with corner coordinates, and the distribution of the generated response map is unknown, which is not very consistent with the corner refinement method.

In this paper, a camera calibration method combining a deep learning corner detection method with the Zhang Zhengyou calibration method is used [14], and improvements are made to this structure. Experimental results show the effectiveness of the proposed method in real-world scenes.

2. Camera Calibration Principles

In the binocular stereo vision system shown in Fig. 1, a point M with coordinates (X, Y, Z) in the world coordinate system will be projected onto the images of the left and right cameras as points m1(uL, vL) and m2(uR, vR), respectively. The baseline b is the line connecting the optical centers of the left and right cameras, and their focal lengths are f. The optical centers of the cameras are OR and OL [13]. The homogeneous coordinates of the pixel coordinate point m(u,

ISSN: 2664-9640

v) are represented as $\tilde{m}(u,v,1)$, and the 3D homogeneous coordinates of the world coordinate point M are $\tilde{M}(X,Y,Z,1)$. According to the pinhole imaging principle, the relationship between the 3D point $\tilde{M}$ and its projection point $\tilde{m}$ is given by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = sA[R,t]\tilde{M} = s\begin{bmatrix} \alpha & \gamma & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

The matrix A is the camera's intrinsic parameter matrix. [R, t] represents the camera's extrinsic parameter matrix, where R is a 3x3 rotation matrix and t is a 3x1 translation vector; 0 and 1 are matrix padding constants. s is the scale factor. In the Zhang Zhengyou camera calibration method, it is assumed during the calibration process that the depth Z = 0, and the intrinsic and extrinsic parameters remain constant. Based on Equation 1, the following can be obtained:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2)$$

The homography matrix H that relates the point \widetilde{M} to its projection on the image m satisfies the relationship $\tilde{M} = sH\tilde{m}$, where H = A[r1 r2 t] = [h1 h2 h3]. This homography matrix contains information about both the intrinsic and extrinsic parameters, and it consists of a total of 8 degrees of freedom. According to the orthogonality of the rotation matrix {r1}^Tr2=0, and the magnitude satisfies $|r1| = |r2| = 1$ | r1 | = | r2 | =1, the intrinsic and extrinsic parameters of the camera are solved.

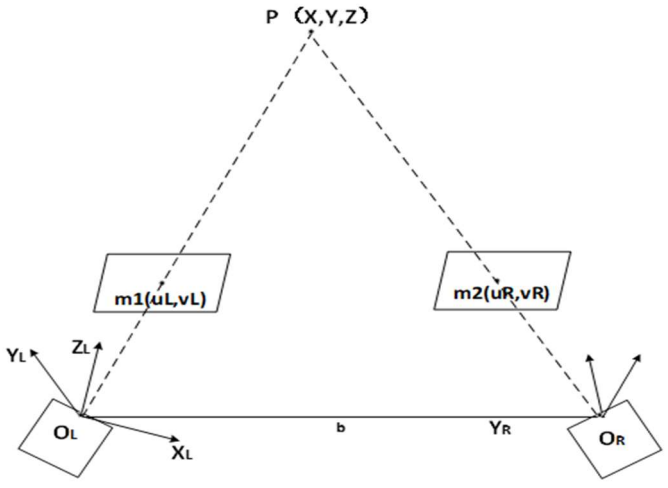

Fig.1 Binocular stereo vision

3. Corner Detection Algorithm based on Deep Learning

Traditional corner detection algorithms are inaccurate, prone to false positives and missed detections, and do not achieve sub-pixel level precision. For high-precision camera calibration, finer sub-pixel level positioning is required. To address these issues, this paper introduces a corner detection method based on deep learning [14], as shown in Fig. 2.

ISSN: 2664-9640

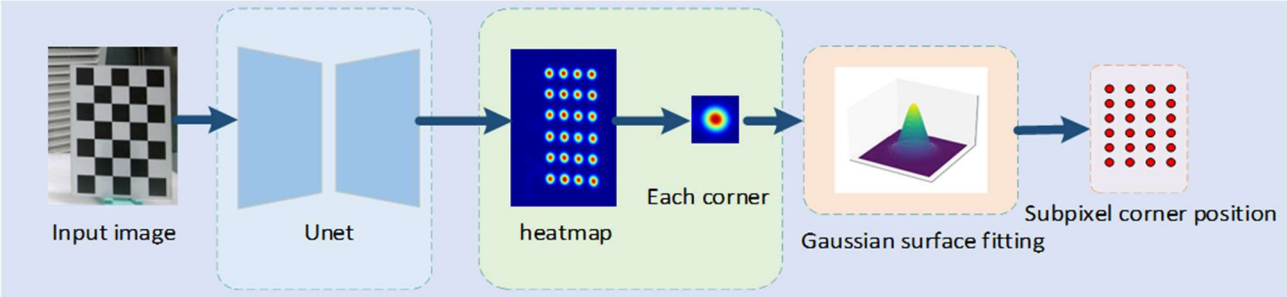

Fig. 2 Corner detection based on UNet

The specific implementation steps are as follows: Use the generated heatmap to train the network model, and its loss function can be defined as:

$$\text{Ldetect} = \iint_{R^2} ||\text{Y1}(x, y - \text{Y}(x, y)|| ^2 dxdy \tag{3}$$

Where Y represents the input heatmap containing the true sub-pixel corner positions, and Y1 denotes the network's output. When a real image is input into the trained model, a heatmap containing the corner positions is generated. A Gaussian surface fitting algorithm is used to extract sub-pixel level corner positions from this heatmap. The formula is as follows.

$$\arg\min_{u,\sigma} ||\text{G}(u, \sigma) - \text{G1}| \tag{4}$$

G1 represents the distribution of each corner point in the heatmap, where μ is the center and σ is the variance.

3.1. Improved UNet Corner Detection Network

The UNet network, with its encoder and decoder structure, is capable of detecting corners on feature maps of different levels. In the contracting path, it captures global context information through pooling operations, and in the expansive path, it recovers detailed information through upsampling and skip connections. This allows the network to understand the position and relationships of corners within images, and the model structure is simple. Based on these advantages, this paper uses the UNet network as the benchmark model for corner detection and innovates on this baseline.Innovations include introducing a spatial multi-scale attention module in the skip connections of UNet. The feature maps of each level are input into the spatial multi-scale attention module, and the results are concatenated with the same level in the upsampling path after passing through the spatial attention multi-scale module. A multi-level attention module is introduced between upsampling and downsampling to reduce false positives and missed detections of chessboard corners. An image reconstruction enhancement module is introduced at the output stage of UNet to reconstruct and enhance the output images. The model structure is shown in the fig. below.

ISSN: 2664-9640

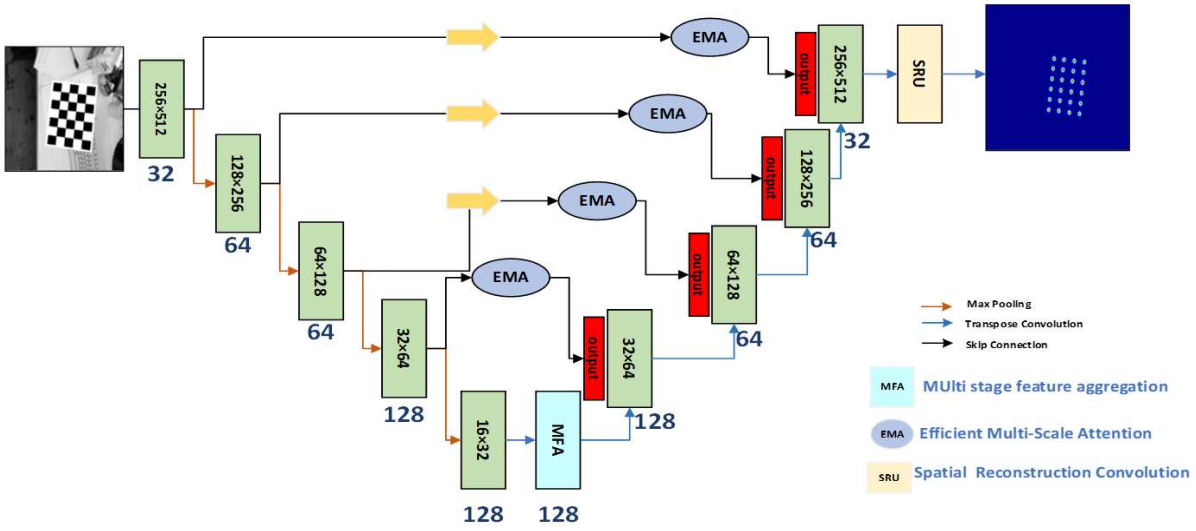

Fig.3 Model overall architecture

3.2. SRU (Spatial Reconstruction Unit) Module

The Image Reconstruction Enhancement Module, as shown in Fig, 3, is designed to reduce redundant features in images, decrease unnecessary computations, enhance the expression of useful features in image sequences, highlight corner features in recognition images, and improve the accuracy of corner detection [15]. The Image Reconstruction Enhancement Module consists of two units: the Spatial Reconstruction Unit and the Multi-scale Feature Fusion Unit.In the Spatial Reconstruction Unit, sequences are reordered based on their average attention weights, with the importance decreasing from F1 to F2, and F1 containing more corner information. However, simply discarding local sequences like F2, which contain less corner feature information, would affect the detection results. These sequences include background information and some auxiliary features that are beneficial for detection. To focus the model more on those regions of higher importance, local sequences of lower importance are fused with preceding local sequences of higher importance to generate new local sequences.The specific implementation involves performing a split operation in the Spatial Reconstruction Unit. The purpose of the split operation is to separate information containing corner feature spatial content from other information that does not contain corner features. The trainable parameter $\gamma \in RC$ in the Group Normalization (GN) layer is used to measure the spatial pixel variance for each batch and channel. The input feature X is standardized by subtracting the mean μ and dividing by the standard deviation σ. This is represented as:

$$X_{\text{out}}=\text{GN}(X)=\gamma\frac{\text{X-}\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{5}$$

Where μ represents the mean, σ denotes the standard deviation, and ε is a constant added for numerical stability during division. The parameters γ and β are trainable affine transformation variables.These parameters enable dynamic adjustment of the activation values across different channel groups, assisting the model in focusing on the features at the locations of the chessboard corners while mitigating the impact of regions that do not contain corner features on the recognition outcomes.The normalized weight $W\gamma \in RC$ is derived from Equation (6), which elucidates the significance of various features.

$$\{W_i\}=\frac{\gamma_i}{\sum_{\text{j=1}}^{C}\gamma_i}\text{, i,j=1,2,...C} \tag{6}$$

ISSN: 2664-9640

The weight values of the feature map reweighted by Wγ are mapped to the range (0, 1) using the sigmoid function, as shown in the following formula:

$$W=\text{Gate}(\text{Sigmoid}(W_{\gamma}(\text{GN}(X)))) \tag{7}$$

Based on threshold gating, these weights are divided into weights W1 that contain more corner feature information and weights W2 that contain less corner feature information. W1 and W2 are used to weight the input features separately, generating two parts of features with strong and weak corner feature expressiveness, respectively. Through cross-reconstruction operations, these two parts of features are effectively combined, reinforcing the information flow between them, in order to generate a more rich and space-saving feature representation.

$$\begin{aligned} X_1^W &= W_1 \otimes X \qquad X_2^W = W_2 \otimes X \\ X &= X_2^W \oplus X_1^W \qquad X_M^W = X_M^W \oplus X_1^W \end{aligned} \tag{8}$$

Where $\otimes$ denotes element-wise multiplication and $\oplus$ represents element-wise addition. Compared to the original sequence, the reconstructed sequence distinguishes between the corner feature areas and other regions based on the proposed average attention weights, and enhances the proportion of corner features in the global features through fusion operations, thereby highlighting the corner features. At the same time, it reduces the impact of interference factors such as noise and background on the entire image area sequence.

The feature map X after spatial group reconstruction still contains some unimportant features such as background regions. Therefore, a multi-scale feature fusion unit is designed to further enhance the chessboard corner features. In the multi-scale feature fusion unit, the input feature map is processed by adaptive dilated convolution layers with dilation factors of 1, 2, and 3, respectively. For the recognition of fine structures like chessboard corners, it is necessary to maintain high resolution; ordinary convolution and pooling would reduce spatial resolution, and the limited receptive field makes it difficult to accurately capture these details. Adaptive dilated convolution, by expanding the receptive field, can better cover and recognize these small targets while retaining more spatial details, thus improving detection accuracy. The formula for obtaining a new feature map by adding and averaging the three sets of feature maps obtained from dilated convolutions is as follows:

$$f = (\varphi^1 3 \times 3(f) + \varphi^2 3 \times 3(f) + \varphi^3 3 \times 3(f)) \tag{9}$$

The obtained feature map is then input into the designed pooling module [16], as shown in Fig, 5. The pooling module consists of three branches. Horizontal and vertical strip pooling operations are utilized to collect remote context information from different spatial dimensions. Strip pooling divides the feature map into several vertical strip-like regions, and each region undergoes independent pooling operations. The purpose of this structure is to capture features along specific directions; corners exhibit directional features along the chessboard pattern. Strip pooling assists the network in focusing on these directional features, thereby enhancing the identification of corners.

ISSN: 2664-9640

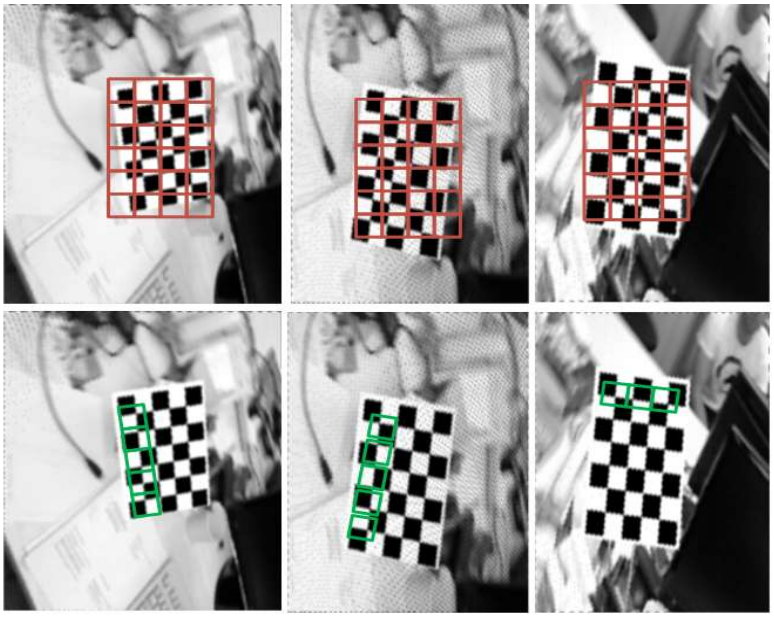

Fig.4 The Working Mechanisms of Strip Pooling and Traditional Pooling in Chessboard Corner Detection

Feature vector X ∈ RH×W×C, where H, W, and C represent the height, width, and number of channels of the feature vector, respectively. In strip pooling, the spatial range that needs to be pooled is (H, 1) and (1, W). Unlike two-dimensional average pooling, the proposed strip pooling computes the average over all feature values within rows or columns.

$$y_i^h = \frac{1}{W} \sum_{0 \le j \le W} X_{\mathrm{i,j}}$$
$$y_j^v = \frac{1}{H} \sum_{0 \le j \le W} X_{\mathrm{i,j}} \tag{10}$$

The third branch is the channel average pooling. Unlike conventional pooling, channel average pooling does not change the spatial dimensions of the feature map, retaining more spatial information, and it captures the interrelations between channels, allowing the model to more accurately locate corners. It computes the average of pixel values within each channel, integrating the spatial information across that channel, and generates a single value for each channel. This value represents the global information on that channel. Subsequently, a 1×1 convolution with a kernel size of 3 is applied to each of the three output feature maps to expand their dimensions. After expansion, the three feature maps have the same dimensions, and the feature map of size H×W is obtained by summing the corresponding positions of the expanded feature maps. The output is then produced by multiplying the result of the 1×1 convolution and sigmoid activation with the corresponding pixels of the original input map. Finally, the outputs of the Spatial Reconstruction Unit and the Multi-scale Feature Fusion Unit are element-wise multiplied to obtain the result of the image reconstruction enhancement.

ISSN: 2664-9640

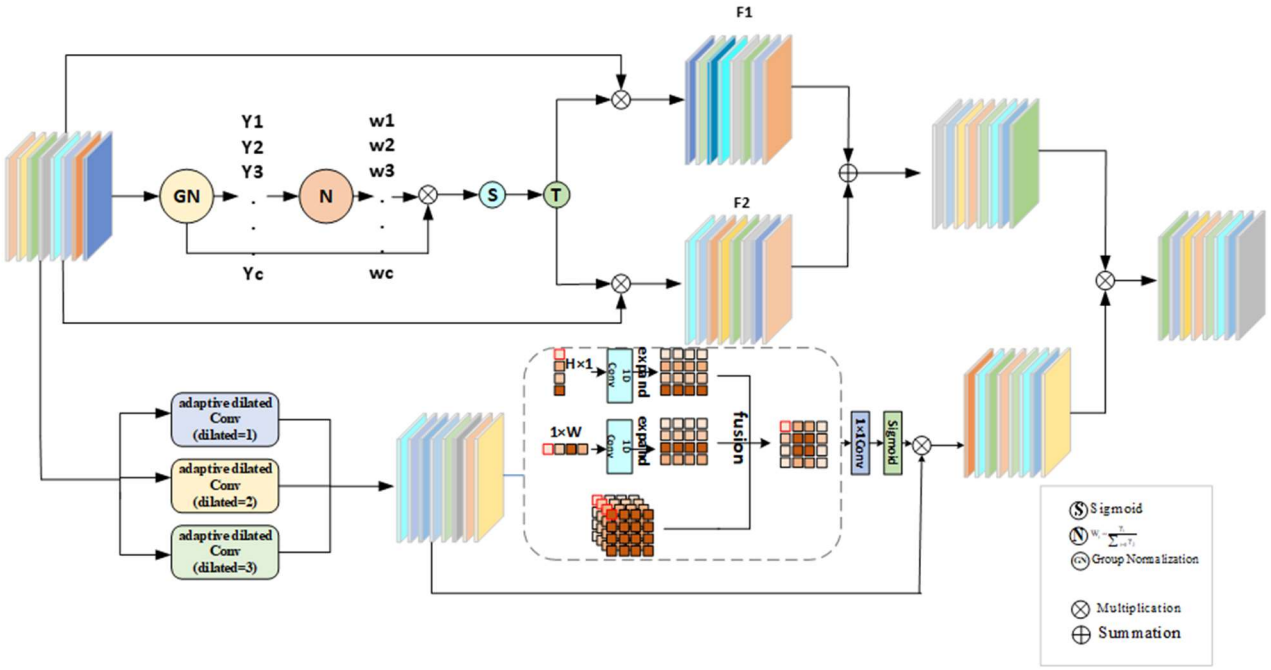

Fig.5 Image reconstruction enhancement module

3.3. MFA (Multilevel Feature Attention) Module

Different levels of features contain different information. By aggregating these features, the model can enhance its representation of corners, helping the network distinguish corners from other types of image features. This paper employs a multi-level attention mechanism that fuses channel and spatial information. The implementation of the MFA (Multilevel Feature Attention) module will be described in detail below. For the input feature map f, three 1×1 convolutions ψq1, ψv1, and ψk1 are used to transform the original feature map into a more compact and informative embedding representation: ψ1q (fh), ψ1v (fl), and ψ1k (fl). The embedding representations of the two level feature maps ψq1(fh) and ψk1(fl) are then computed to form a similarity matrix, which is then normalized through a softmax function to obtain a channel-wise similarity matrix Mc.

$$Mc = Fsoftmax(\psi q1(fh) \times \psi k1(fl)) \tag{11}$$

After calculating the channel-wise similarity, the feature map ψv1(fl) is multiplied by the matrix Mc through matrix multiplication to recover the complete channel dimension information. Then, a 1×1 convolutional layer ωc is used to adjust the size of the feature map to match the dimensions of feature map fh, completing the multi-level feature aggregation across channels. This results in a new feature map fch after feature aggregation, expressed as:

$$fch = \omega c(\psi v1(fl) \times Mc) + fh \tag{12}$$

The MFA module further enhances feature expression by performing spatial feature aggregation operations. Similar to the method previously used for channel feature aggregation, the attention weights are fused with spatial-dimensional information. This fusion is achieved through two 1×1 convolutional layers ωs and ψv2, along with the pre-computed spatial similarity matrix Ms, to spatially integrate the feature maps fl and the previously aggregated fch. Ultimately, this process generates the complete output feature representation fsh of the MFA module, with its mathematical expression being:

ISSN: 2664-9640

$$fch = \omega s(\psi v2(fl) \times Mc) + fh \quad (13)$$

This attention mechanism consists of two levels: low-level and high-level. In the low-level attention part, the model uses self-attention mechanisms to encode the input sequence, obtaining the representation vectors for each position and the local relationships within the input sequence. In the high-level attention part, the model receives the encoded representation vectors from the low level as input and evaluates and calculates the importance weights for each position through the attention mechanism. Based on these weights, the low-level representations are weighted and fused. This multi-level attention-guided feature aggregation provides an effective way for the model to explore and utilize the channel and spatial feature representations at different levels, allowing the model to more accurately identify and locate chessboard corners, enhance its ability to differentiate various types of feature points, and thus effectively reduce false positives and false negatives in corner detection.

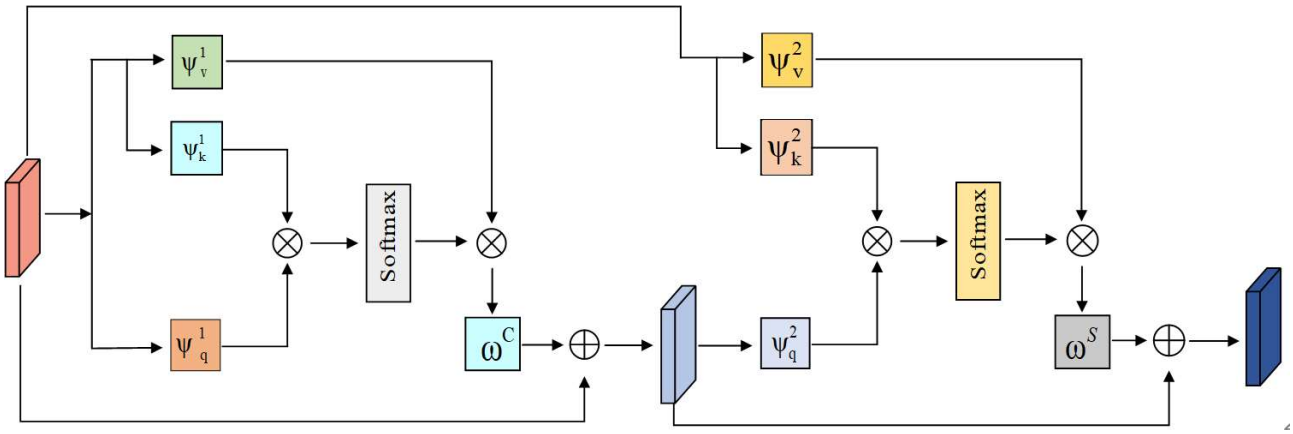

Fig. 6 Multilevel attention module

3.4. Spatial Attention Module

In the task of detecting chessboard corners, it is necessary to accurately identify the features of the corners. The spatial attention mechanism focuses on spatial information in the image, helping the model to identify the regions of interest in the image and adjust the model's attention accordingly. In this way, the spatial attention mechanism can increase the model's focus on the key areas in the image. This paper compares the CBAM (Convolutional Block Attention Module) spatial attention mechanism with a new EMA (Efficient Multi-Scale Attention) spatial attention mechanism, and analyzes the results in ablation experiments, ultimately selecting EMA spatial attention as the method in this paper.

The CBAM attention mechanism is divided into two parts: the channel attention module and the spatial attention module. The channel attention module performs max pooling and average pooling on the input feature map along the height and width dimensions, respectively, to obtain two two-dimensional feature maps (H×W×1). The two feature maps after pooling are processed by a convolutional layer with shared weights, and the output of the convolutional layer is passed through a Sigmoid activation function to generate two weight matrices, corresponding to the results of max pooling and average pooling, respectively. Finally, the two weight matrices are added together to obtain a channel attention weight matrix. In the spatial attention module, the output of the channel attention module is used as input, and similar to the channel attention module, the input feature map is processed along the channel dimension with max pooling and average pooling, resulting in two two-dimensional feature maps (H×W×1). The results of max pooling and average pooling are concatenated along the channel dimension to obtain a feature map with a dimension of H×W×2. The concatenated feature map is then processed by a

ISSN: 2664-9640

convolutional layer, and the output of the convolutional layer is passed through a Sigmoid activation function to generate a spatial attention weight matrix, representing the importance of each spatial position. Finally, the spatial attention weight matrix is multiplied by the output feature map of the channel attention module to obtain the final weighted feature map. The CBAM spatial attention module is illustrated as follows.

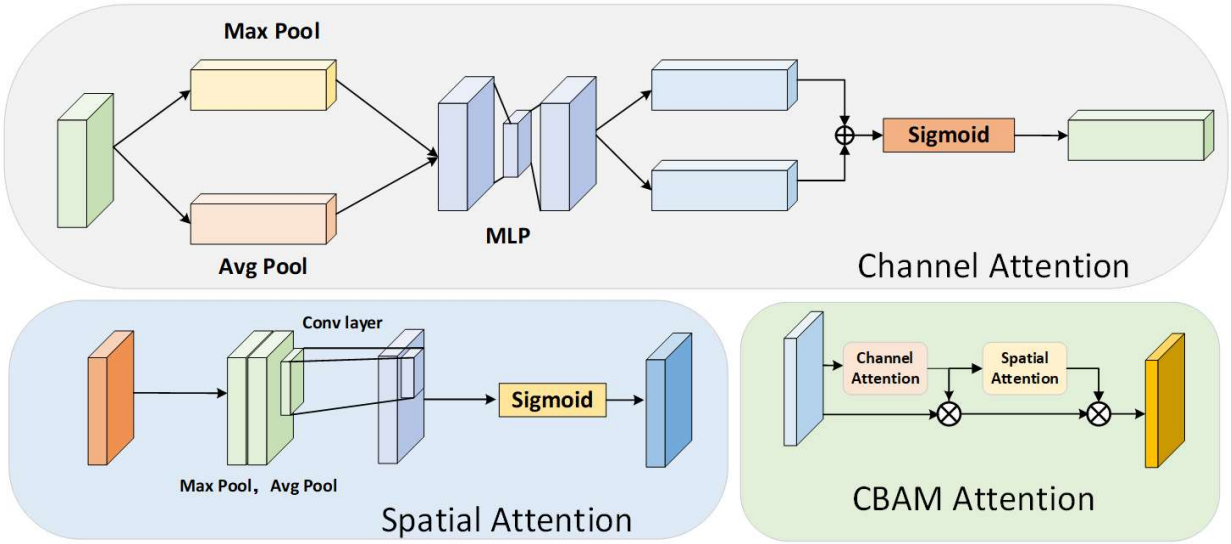

Fig. 7 CBAM attention module

Traditional spatial attention establishes relationships between channels through dimensionality reduction. However, during the process of establishing channel relationships through dimensionality reduction, some critical information between channels may be lost, weakening the model's deep understanding and expression capabilities for chessboard features. To address this issue, this paper introduces the EMA (Efficient Multi-Scale Attention) module, which can effectively preserve and utilize rich information between channels while reducing computational complexity. Below, the specific implementation steps of the EMA spatial attention module will be described in detail.

In the EMA module, for the input feature map $X \in R^{C\times H\times W}$, it is divided into g groups of sub-features along the channel dimension to obtain $X \in R^{C/G\times H\times W}$. Then, through three parallel branches, for the first two branches, horizontal average pooling along the spatial dimension is performed to obtain $X_{avg}^{X} \in R^{C\times H\times W}$, and vertical average pooling is performed to obtain $X_{avg}^{Y} \in R^{C\times H\times W}$. This operation can capture long-distance dependencies in the horizontal direction and preserve the vertical positional relationships, helping the model to more accurately locate corner features. The outputs of the first two branches are then concatenated along the spatial dimension to obtain $X_{avg}^{XY} \in R^{C\times H\times W}$.

$$X_{avg}^{XY} = Concat\left(X_{avg}^{X}, X_{avg}^{Y}\right) \tag{14}$$

The obtained feature is then input into a 1×1 convolutional layer to obtain the intermediate feature map $F=Conv(X_{avg}^{XY})$. The F is split along the spatial direction into two components, and then reshape operations are performed on the two components. Sigmoid gating units are used to obtain weights Wx and Wy for the two components, respectively. Finally, the X is adjusted using Wx and Wy.

ISSN: 2664-9640

$$Xc = X \odot Wx \odot Wy \tag{15}$$

In the cross-spatial information aggregation part, we introduce two feature tensors: one is the output of the 1×1 convolution branch, and the other is the output of the 3×3 convolution branch. A 2D global average pooling layer is used to encode global spatial information from the output of the 1×1 branch to obtain the channel descriptor X1. X1 is then multiplied with the output of the 3×3 branch Xs to weight and sum all channel features at each position, resulting in the global spatial attention representation at the 1×1 scale. Similar operations are performed using a 2D average pooling layer on the output of the 3×3 branch to encode global spatial information and obtain Xs. A Softmax activation is applied to obtain the normalized channel descriptor X2, which is reshaped to have the same shape as X2. The channel descriptor X2 is then multiplied with the output of the 1×1 branch Xc, weighting and summing all channel features at each position to obtain the global spatial attention representation at the 3×3 scale. The two global spatial attentions preserve spatial position information at different scales. Finally, the two spatial attentions are aggregated, and a Sigmoid gating function is applied to obtain the weight W. The input feature X is then recalibrated using the weight W to obtain the output.

$$out = X \odot W \tag{16}$$

Finally, through a reshape operation, the output is restored to the same shape as the input, yielding the final output of the EMA module.

The EMA module ensures the even distribution of spatial semantic features among the subgroups by recombining a portion of the channels into the batch dimension and segmenting the channel dimension into multiple sub-feature groups. This parallel processing approach captures structural information at different spatial scales. By combining both 1x1 and 3x3 convolution branches, the 1x1 branch provides rapid response with lower computational overhead, while the 3x3 branch captures local context information. Unlike the traditional CBAM spatial attention that simply uses an average method to aggregate attention weights, the EMA module uses matrix dot product operations to capture pixel-level pairwise relationships, emphasizing the impact of global context on each pixel. This design not only enhances the recognition ability of corner features but also optimizes the process of attention allocation at the pixel level, thereby improving the model's understanding of the input data.

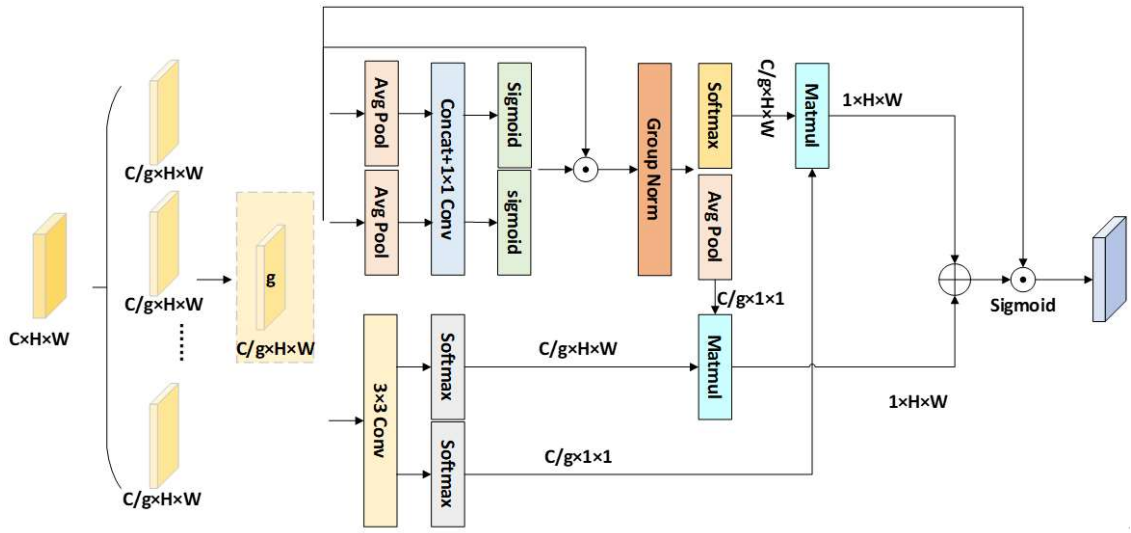

Fig. 8 Multi-scale spatial attention module

ISSN: 2664-9640

4. Experimental Results and Analysis

The network model in this paper is implemented using the PyTorch deep learning framework. The hardware configuration includes a V100-32GB GPU with 32 GB of memory. The software configuration is based on the cuda11.3 compute architecture version. Additionally, an Anaconda virtual environment is set up, with important dependencies including python3.8, pytorch1.10.0, and Numpy1.24.3. Ablation experiments are conducted to demonstrate the impact of the innovations in each module on corner detection performance. Comparisons with the original model algorithm and other model algorithms are made to prove the effectiveness of the algorithm proposed in this paper.

4.1. Experimental Details

Randomly generate a large number of virtual chessboards with a resolution of 480×480, and use the TUM RGBD dataset to generate fake backgrounds for the chessboards to enhance the data. Each generated chessboard corner has a Gaussian distribution representation, with the center of the Gaussian distribution corresponding to the annotated sub-pixel position. Training data consists of the generated virtual chessboard images and heatmaps containing the true sub-pixel positions of the chessboard corners. The network model is trained on this data. Training is performed for 1000 epochs on the randomly generated dataset, with an initial learning rate of 0.001, which is halved after the 600th epoch.

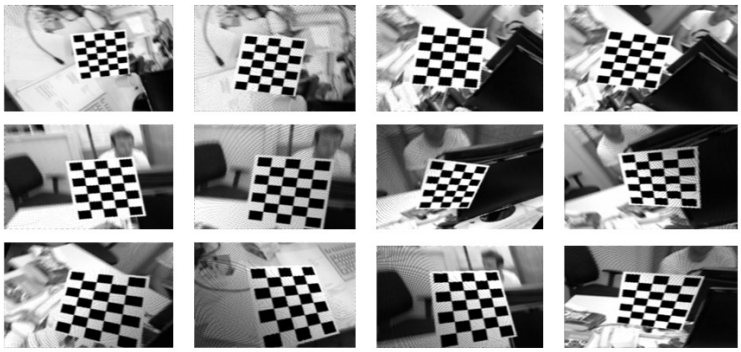

Fig. 9 A virtual checkerboard image is generated

4.2. Corner Detection Ablation Experiments

Table 1. The results of corner detection ablation experiment

Category	scale				
SRU		√			√
MFA			√		√
EMA				√	√
Corner error =0.0–0.2 pixel	0.0416	0.0638	0.0527	0.0555	0.0972
Corner error =0.2–0.4 pixel	0.1944	0.3194	0.2444	0.2583	0.3444
Corner error 0.4–0.6 pixel	0.3361	0.3500	0.3416	0.3416	0.3361
Corner error =0.6–0.8 pixel	0.3250	0.2527	0.3027	0.2915	0.1667
Corner error =0.8–0.9 pixel	0.0722	0.0138	0.0444	0.0472	0.0556
Corner error >0.9 pixel	0.0305	0.0000	0.0138	0.0055	0.0000
Ed	0.55	0.46	0.51	0.48	0.43

In this paper, the accuracy of the corner detection method is tested on the generated virtual chessboard data. Fifteen generated virtual chessboards are selected for corner detection in the test images, each chessboard has a size of 6×4 and contains 24 corners, a total of 360 corners.

ISSN: 2664-9640

All corners in these 15 images can be detected, The corner position is denoted as (x', y'), and the true corner position is also denoted as (x, y). For each pair of corners, the detection error $= \sqrt{(y - y')2 + (x - x')2}$. for each pair of corners is calculated and accumulated to the total error. After processing all corners, the total error is divided by the number of corners to obtain the average pixel error. The Corner error in different detection error intervals is the ratio of the total number of corners in that interval to the total number of corners detected by the model. The ablation experiment results are shown in Table 1.

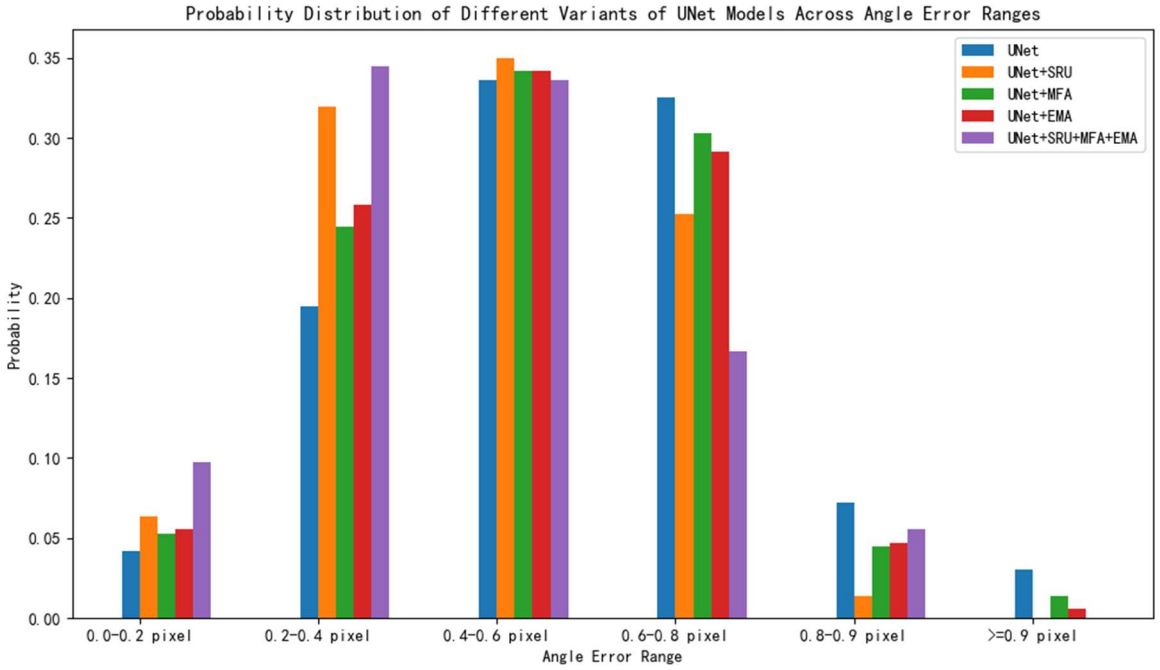

Fig. 10 Corner error probability distributions for different UNet variants

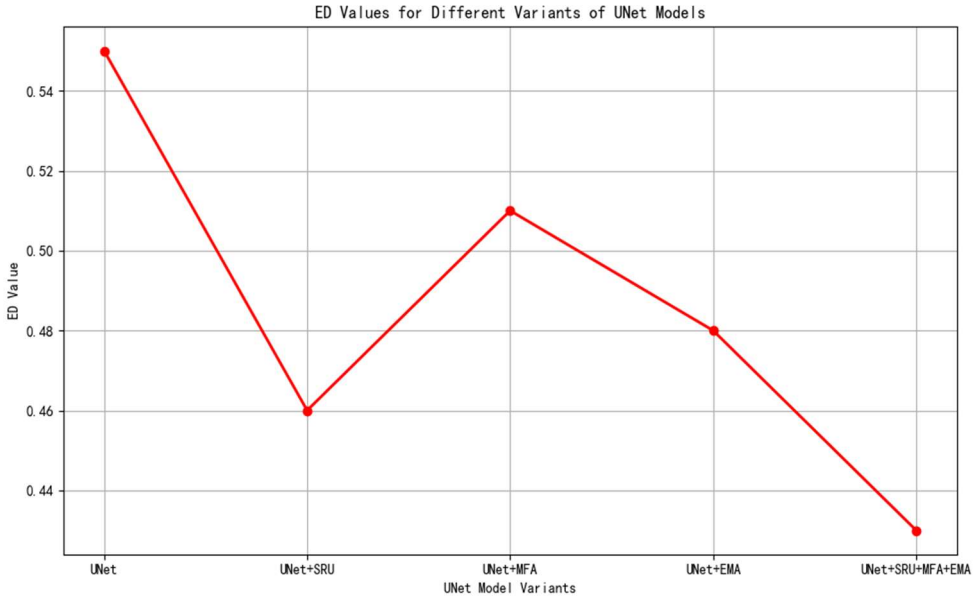

Fig. 11 average pixel error of different methods

Using UNet [19] as the baseline network, ablation experiments are conducted on randomly generated virtual chessboards to prove the effectiveness of each module. There are a total of 360 corners in 15 virtual chessboards. Using UNet for corner detection results in an average

ISSN: 2664-9640

pixel error of 0.55. After introducing the Image Reconstruction Enhancement module, the average detection error is reduced to 0.46, which is lower than the baseline network UNet; the proportion of corners with error intervals between 0-0.2 pixels is greater than that detected by UNet, proving the effectiveness of the Image Reconstruction Enhancement module.Introducing the MFA module reduces the average detection error to 0.51, and the proportion of corners with error intervals between 0-0.2 pixels is greater than that detected by UNet, proving the effectiveness of the MFA module. Similarly, introducing the EMA module reduces the average detection error to 0.48, and the proportion of corners with error intervals between 0-0.2 pixels is greater than that detected by UNet, further proving the effectiveness of the EMA module.When integrating these modules into the model, the average detection error is reduced to 0.43, which is lower than the baseline network UNet; the proportion of corners with error intervals between 0-0.2 pixels is higher than that after introducing each module, indicating that the proposed model performs better with the combined effect of these modules, thereby proving the effectiveness of the modules proposed in this paper for corner detection.

To illustrate the role of each module in reducing false positives and false negatives in chessboard corner detection, partial corner detection images are provided on the generated virtual dataset in Fig. 12.

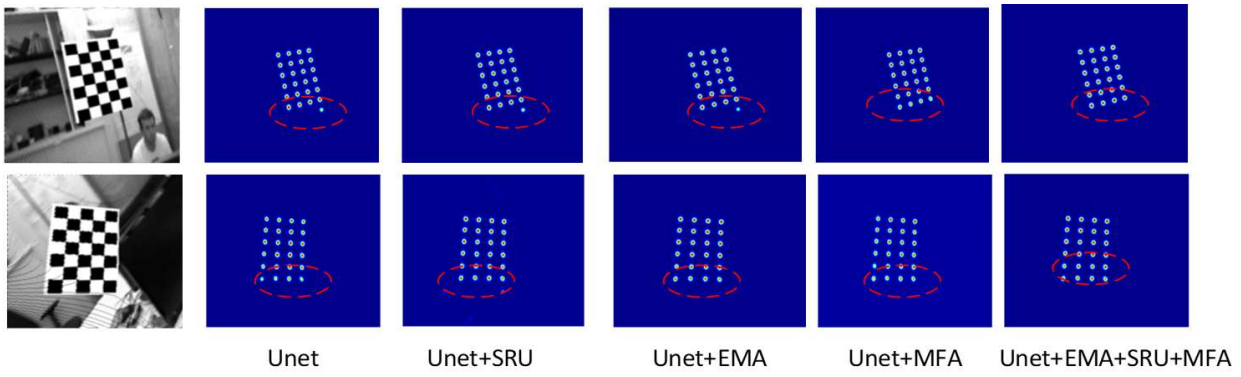

Fig. 12 Virtual Chessboard Image Corner Detection

We selected two sets of images from the virtual chessboard test set for analysis. In the first set of images, when using the UNet baseline model, the corners at the left and right edges of the last row of the chessboard cannot be detected. After introducing the Image Reconstruction Enhancement module and the cross-spatial multi-scale attention, the model's focus on corner features has improved, and all corners can be detected, but there are still issues with false positives. After introducing the multi-level attention module, the detection rate of corners is increased compared to the baseline model UNet, but some corners are still not detected.

In the second set of images, the model fails to distinguish points with similar features to the chessboard corners in the background. After introducing the Image Reconstruction Enhancement module and the cross-spatial multi-scale attention, false positive corners still exist. However, after introducing the multi-level attention module, the model can better differentiate the features of chessboard corners from other points, eliminating the model's false positive issues. With the combined effect of all modules, the model's problems with false positives and false negatives are largely resolved.

Since camera calibration requires corner detection on real chessboard images, this paper focuses on the effect of corner detection on real chessboard images and provides partial experimental effect images on the real dataset for analysis to demonstrate the effectiveness of each module, as shown in Fig. 13.

ISSN: 2664-9640

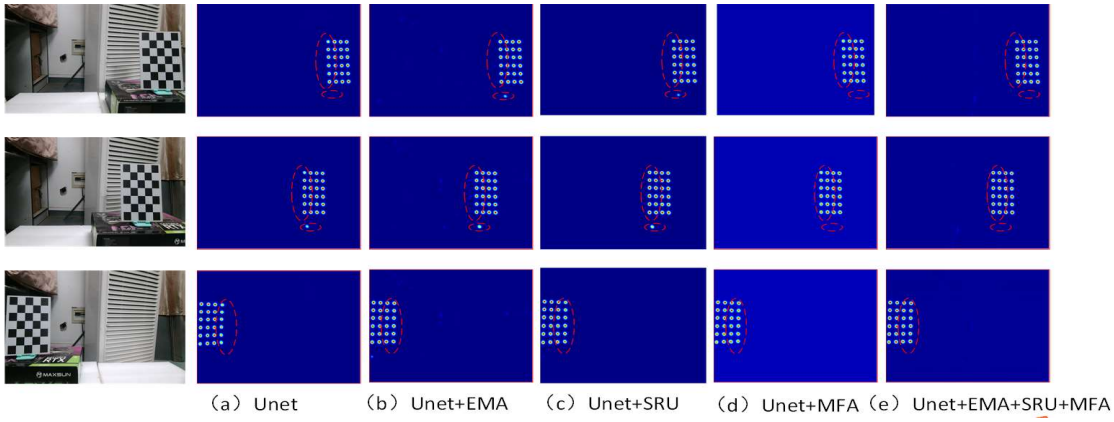

Fig. 13 Real Chessboard Corner Detection

From the figure, it can be observed that when using the UNet network for corner detection, individual corners at the left or right edges may not be detected, and there is a possibility of false detection. After introducing the cross-spatial multi-scale module and the image reconstruction enhancement module, points at the left and right edges can be accurately detected, but there are still issues with false detection. This indicates that the module can enhance the model's sensitivity to corner features and improve the accuracy of corner detection. From the small light spots circled in the figure, it can be seen that the model may have false detection issues when the image background and chessboard texture are similar, or when the pattern is similar. After introducing the image reconstruction enhancement module and the spatial attention module, the light spots do not disappear. When this background feature is closer to the corner feature of the chessboard, it will be mistakenly considered a corner during the detection process. After introducing the multi-level attention module, the model can more accurately distinguish between chessboard corners and other similar corner features, as evidenced by the disappearance of the detected light spots in the figure. When combining all three modules, all corners can be accurately detected, the corner positions are closer to the true corner positions, and there are no issues with false detection of corners. This proves the effectiveness of each module, and the combined effect of the modules is better.

4.3. Ablation Experiment Analysis

The task of detecting chessboard corners requires accurate detection of the corner features. The spatial attention mechanism focuses on spatial information in the image, i.e., the relationships between different pixels in the image. It can help the model identify areas of interest in the image. This paper attempts three different spatial attention mechanisms: the first is the CBAM (Convolutional Block Attention Module) spatial attention module, the second is the EMA (Efficient Multi-Scale Attention) spatial attention module, and the third is to serialize EMA and CBAM spatial attention. The specific implementation steps are to introduce spatial attention in the skip connection part of UNet. The feature map of each layer is used as the input of the spatial attention module, and after passing through the spatial attention module, the result is concatenated with the same layer in the upsampling path. The ablation results using the three different spatial attentions are shown in the table below.

After introducing spatial attention, the corner detection rate is improved compared to the baseline model UNet, and the average pixel error is lower. However, by reducing the dimension to establish relationships between channels, some useful information between channels may be reduced, leading to a decrease in the model's understanding and expression capabilities of input features. This can destroy the rich structures within the feature map and the correlations between channels, weakening the model's ability to learn complex patterns and weakening the features of chessboard corners. The spatial multi-scale attention module can effectively

ISSN: 2664-9640

preserve and utilize the rich information between channels. After introducing the EMA module, the corner detection rate is higher than that of the model after introducing the CBAM module, and the average pixel error is lower. When both are serialized and connected to the model, the corner detection rate is slightly improved compared to the baseline model, but it is lower than that of each module individually, and the average pixel error is lower than that of the baseline model UNet. The effect of using both spatial attentions in series is worse. The model becomes overly complex when both spatial attentions are used in series, which can lead to a decrease in model performance in certain cases. Therefore, the spatial multi-scale attention is ultimately chosen as the method used in this paper.

Table 2. The results of corner detection ablation experiment

Category	Module Ablation			
CBAM		√		√
EMA			√	√
Corner error =0.0–0.2 pixel	0.0416	0.0444	0.0555	0.0361
Corner error =0.2–0.4 pixel	0.1944	0.2055	0.2583	0.1861
Corner error 0.4–0.6 pixel	0.3361	0.3472	0.3416	0.3194
Corner error =0.6–0.8 pixel	0.3250	0.3305	0.2915	0.3333
Corner error =0.8–0.9 pixel	0.0722	0.0527	0.0472	0.0833
Corner error >0.9 pixel	0.0305	0.0194	0.0055	0.0416
Ed	0.55	0.51	0.48	0.58

4.4. Corner Detection Comparison Experiments

The results of the comparison experiments among different corner detection algorithms are shown in Table 3.

Table 3. Comparison of experimental results of corner detection

Method	Libcb[20]	OpenCV[21]	Harris[22][22]	UNet	Ours
Ed	2.52	2.23	0.64	0.55	0.43

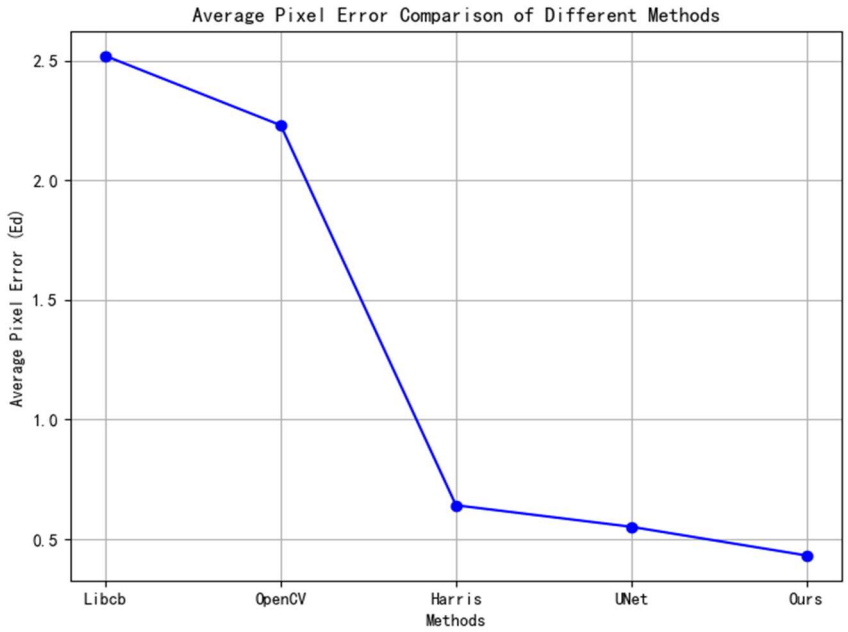

Fig.14 Comparison of corner detection by different methods

ISSN: 2664-9640

As can be seen from Table 2, the network model proposed in this paper has an average detection error of 43%, which is lower than other corner detection algorithms. Compared with traditional sub-pixel corner detection algorithms, the proposed algorithm has the lowest average pixel error.

4.5. Stereo Camera Calibration Experiments

The chessboard corner detection is implemented using the Harris corner detection algorithm and the corner detection algorithm proposed in this paper. According to the Zhang Zhengyou camera calibration method, the parameters of the left and right cameras as well as the camera's extrinsic parameters are obtained. The camera intrinsic and extrinsic parameters obtained from the Harris corner detection algorithm and the proposed method are shown in Table 4. fx and fy represent the normalized focal lengths along the x-axis and y-axis of the image, respectively, in units of pixels. k1 and k2 denote the radial distortion caused by lens manufacturing defects or non-ideal shapes. Tx, Ty, Tz are the translation vector parameters of the camera relative to the left camera. Cx and Cy represent the horizontal and vertical coordinates of the origin of the image coordinate system on the image plane, respectively. The rotation vectors Rx, Ry, Rz represent the rotation angles around the x, y, and z axes, respectively. These rotation matrices are obtained by associating a rotation axis and rotation angle using the Rodrigues formula.

Table 4. Camera Parameter Calibration Results

argument	Harris	Textual method
Left camera focal length (fx, fy)	(1047.2652,1048.1153)	(1049.6605, 1049.4557)
Right camera focal length (fx, fy)	(1049.4207,1049.4507)	(1048.5490, 1048.4798)
Left camera main point (Cx, Cy)	(944.2135,525.2804)	(954.8329, 526.4357)
Right camera main point (Cx, Cy)	(954.3035,536.1620)	(955.1391, 534.6007)
Left camera distortion	(k1=0.0730, k2=-0.1078)	(K1=0.0760, K2=-0.1052)
Right camera distortion	(k1=0.0740, k2=-0.1103)	(K1=0.0773, K2=-0.1062)
Rotation vector	(Rx=0.0114, Ry=-0.0116, Ry=0.0068	(Rx=0.0128, Ry=-0.0109, Ry=0.0077)
Tx, Ty, Tz	(39.6477, -0.3189, -1.6705)	(39.9136, -0.3252, -1.685)

The calibration results of the Harris corner detection algorithm and the proposed corner detection method are evaluated using the reprojection method. The calculated reprojection errors are 0.03 pixels for the Harris corner detection algorithm and 0.02 pixels for the proposed method. Compared with traditional sub-pixel corner detection algorithms, the proposed corner detection algorithm has higher camera calibration accuracy.

5. Summary

This paper designs a binocular camera calibration method based on deep learning corner detection. UNet is used as the baseline network for corner detection, and innovations are made on top of this model. The results of virtual chessboard image corner detection show that the improved model has an average detection error of 43%, and the detection accuracy is significantly improved compared to the baseline model. It also achieves good results compared with traditional corner detection algorithms, and experiments on real chessboards also obtain good detection results. Finally, the improved corner detection algorithm is used for binocular calibration, with a reprojection error lower than the reprojection error obtained by the traditional sub-pixel corner detection method Harris, improving the accuracy of camera calibration. The next step is to try using other models for corner detection to further improve

ISSN: 2664-9640

the accuracy of corner detection, and apply this algorithm to practical binocular vision 3D reconstruction tasks to verify its effectiveness and practicality.

Acknowledgments

The authors gratefully acknowledge the financial support from xxx funds.

References

[1] Zheng Taiqiong, Huang Shuai, Li Yongfu, et al. “A Survey of Key Techniques in Vision-based 3D Reconstruction.” Journal of Automation Science and Engineering, 2020, 46(04): 631-652. DOI: 10.16383/j.aas.2017.c170502.

[2] The key technologies of binocular vision: a review. Huang Ji-yuan, Li Min, Xie Bing-bing, et al. Journal of Manufacturing Automation, 2023, 45 (05): 166-171.

[3] Le Chong, Zhang Jianxun, Liao Ning. Overview of Traditional Camera Calibration Algorithms [J]. Scientific Consultation (Science & Management), 2018, (01): 38-39.

[4] Meng Xiaoqiao, Hu Zhengyi. Research and progress on camera self-calibration methods [J]. Acta Automatica Sinica, 2003, 29(1):15. DOI:CNKI:SUN:MOTO.0.2003-01-014.Y.I.Abdel-Aziz, H.M.Karara, Michael Hauck. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. Photogrammetric Engineering & Remote Sensing, 2015,81(2): 103~107.

[5] Tsai R Y.An efficient and accurate camera calibration technique for 3D machine vision.Proc CV PR,1986: 364~374 [38] Zhang Zhengyou. A flexible new technique for camera calibration. IEEE Transactions on PatternAnalysis and Machine Intelligence,2000,22(11):1330~1334.

[6] Zhang Zhengyou. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence,2000,22(11): 1330~1334.

[7] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[EB/OL]. (2015-11-19) [2022-09- 08].

[8] S. Donne, J. De Vylder, B. Goossens, and W. Philips, “MATE: Machine Learning for Adaptive Calibration Template Detection,” Sensors, vol. 16, no. 11, pp. 1858–17,Nov. 2016.

[9] B. Chen, C. Xiong, and Q. Zhang, “CCDN: Checkerboard Corner Detection Network for Robust Camera Calibration,” in Intelligent Robotics and Applications. Cham:Springer, Cham, Aug. 2018, pp. 324–334.

[10] H. Wu and Y. Wan, “A highly accurate and robust deep checkerboard corner detector,” Electronics Letters, vol. 57.

[11] no. 8, pp. 317–320, 2021.

[12] B. Chen, Y. Liu, and C. Xiong, “Automatic checkerboard detection for robust camera calibration,” in 2021 IEEE International Conference on Multimedia and Expo (ICME).IEEE, 2021, pp. 1–6.Xie Q W, Long Q, Mita S. Integration of optical flow and Multi-Path-Viterbi algorithm for stereo vision[J]. International Journal of Wavelets, Multiresolution and Information Processing, 2017, 15(3): 1750022.

[13] Sturm P, Ramalingam S. A generic concept for camera calibration[C]//Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part II 8. Springer Berlin Heidelberg, 2004: 1-13.

[14] Zhang Y, Zhao X, Qian D. Learning-based framework for camera calibration with distortion correction and high precision feature detection[J]. arxiv preprint arxiv:2202.00158, 2022.

[15] Li J, Wen Y, He L. Scconv: spatial and channel reconstruction convolution for feature redundancy[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 6153-6162.

ISSN: 2664-9640

[16] Hou Q, Zhang L, Cheng M M, et al. Strip pooling: Rethinking spatial pooling for scene parsing[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 4003-4012.

[17] Zhang Y, Wang H. Diverse embedding expansion network and low-light cross-modality benchmark for visible-infrared person re-identification[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 2153-2162.

[18] Ouyang D, He S, Zhang G, et al. Efficient multi-scale attention module with cross-spatial learning[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.

[19] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015: 234-241.

[20] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

[21] A. Geiger, F. Moosmann, O. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in International Conference on Robotics and Automation (ICRA), St. Paul, USA, May 2012.

[22] Harris C, Stephens M. A combined corner and edge detector[C]//Alvey vision conference. 1988, 15(50): 10-5244.