Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4611508, 16 pages
https://doi.org/10.1155/2022/4611508

Research Article

A Target Corner Detection Algorithm Based on the Fusion of FAST and Harris

Yifan Wang, Yang Li, Jiaqi Wang, Haofeng Lv, and Zeli Yang

School of Electric and Information Engineering, Changchun University of Science and Technology, Changchun 130022, China

Correspondence should be addressed to Yang Li; lyang@cust.edu.cn

Received 20 December 2021; Revised 24 February 2022; Accepted 5 March 2022; Published 2 April 2022

Academic Editor: Rafal Zdunek

Corner detection is a common method to obtain image features, and the detection effect influences the performance of matching and tracking directly. A FAST-Harris fusion corner detection algorithm is proposed to improve the shortcomings of the Harris algorithm, such as the low detection accuracy and low positioning accuracy, and a corner detection fusion model is established. First, the detected target image is padded, and then the FAST algorithm is used with a 25% reduced contrast points to achieve fast capture roughly; in this way, a candidate corner set is obtained. Then, screening the candidate corner is set one by one by calculating the response function of the Harris with Scharr operator to achieve capture accurately. Finally, the real corners are obtained using SAD for nonmaximum suppression. The positioning error, error detection rate, robustness, and running time of corner detection are obtained by the PyCharm platform. Compared with Harris, the error detection rate and localization error of the algorithm are reduced by 16.89% and 42.04%, respectively. Compared with 8 popular corner detection algorithms, the error detection rate and localization error of the algorithm in this paper are the lowest, which are 24.60% and 1.42 pixels. The robust performance in lossy JPEG compression is the best, with 17.37% shorter running time than Harris algorithm. The method in this paper can be used in scenarios such as autonomous driving and image search services.

1. Introduction

Corner detection is a method used to obtain image features in computer vision which is widely used in motion estimation [1], image matching [2–6], image visual processing [7], visual tracking [8–10], and 3D scene reconstruction, etc. [11–15], mainly divided into edge-based corner detection algorithm [16] and gray-based corner detection algorithm [17]. The corner detection algorithm based on edge has prime requirements for image segmentation and edge detection, and the algorithm is complex and the steps are cumbersome. The gray-based corner detection algorithm calculates the curvature according to the gradient change of the local grayscale of the image and does not need to segment the image and detect the edge in advance.

Harris algorithm [18] is proposed on the basis of gray corner detection algorithm Moravec [19]. It uses the Taylor series expansion to extend the four moving directions to any direction, so that Harris algorithm has better robustness and detection accuracy, but the corner detection rate decreases. In [20], Sobel edge detection is proposed to extract the alternative corners. When nonmaximum suppression is carried out, the rectangular template is changed to the circular template to improve the detection speed. The computational performance and repeatability of [21] are better than [20, 22–25]. The eigenvalues of the matrix are directly calculated by the improved Harris algorithm of Shi-Tomasi, the smaller eigenvalues are compared with the threshold value, and the strong feature point is greater than the threshold value, which effectively improves the detection accuracy [26]. In [27] and [28], the linear combination coefficient and subpixel Harris algorithm are introduced, respectively. Only the corners near the enhanced edge and the parameters of the rectangular template are normalized to improve the accuracy of corner detection. Lowe proposed a Scale Invariant Feature Transform (SIFT) algorithm [29] with rotation invariance, scale invariance, and light intensity invariance. Based on [29], Bay et al. in [29] proposed an accelerated robust feature (Speeded Up Robust Feature, SURF) algorithm [30] to improve the speed of the algorithm while ensuring the invariance of scale and affine transformation. In AKAZE [31], Fast Explicit Diffusion (FED) is added to the pyramid framework to quickly construct a nonlinear space. In the corner detection algorithm of subpixel accuracy [32], one is to get the coordinate value of subpixel accuracy by minimizing the iterative method of error function. The Features from Accelerated Segment Test (FAST) algorithm [33] is the most efficient algorithm in the grayscale corner detection, but there are many redundant corner points detected and there is an image edge missed detection. In order to solve the shortcomings of the FAST algorithm, Rublee et al. proposed the Oriented FAST and Rotated BRIEF (ORB) algorithm [34] which has rotation invariance, fast calculation speed, and antinoise. Researchers usually combine the FAST algorithm with the traditional corner detection algorithm [35–37] or improve the FAST algorithm [38] with the back propagation neural network. Finally, the nonmaximum suppression method is used to screen corners [39] to solve the problem of detecting redundant corners.

By means of the above analysis the Harris algorithm has high detection accuracy, while having a large amount of computation, and the corner detection efficiency is low. FAST algorithm has fast corner detection speed, but it detects more redundant corners. In order to achieve high positioning accuracy, detection accuracy, and robustness, this paper proposes a corner detection algorithm that combines FAST and Harris (F–H). The improved FAST algorithm is used to quickly capture the corner position, and then the improved Harris algorithm is used to finely screen the corner points. The mathematical model is established by gradually narrowing the corner capture range. First, the target image is padded with constants to solve the problem of image edge missed detection. Then, the corner points are quickly and roughly captured by using a 7 * 7 rectangular window for the target image with the advantage of high detection efficiency of the coarse capture FAST algorithm. Using the advantage of high detection accuracy of Harris algorithm, a 3 * 3 rectangular window is used to filter the candidate corner set. Finally, Sum of Absolute Difference (SAD) is used for nonmaximum suppression to reduce redundant corners and get real corners. The simulation experiment is completed based on PyCharm-python3.8 platforms. The results show that the proposed algorithm has better performance in corner missed detection rate and positioning error than eight advanced algorithms and has obvious advantages in detection speed compared with Harris algorithm.

The contributions of the paper are summarized briefly as follows: a fusion corner detection model of FAST and Harris is established. First, the number of comparisons between the center point and the discrete points on the circumference is reduced by 25%, and a 7 * 7 rectangular window is used to capture the corner points roughly to obtain a set of candidate corner points. Then, the Scharr operator is used to replace the Sobel operator in the Harris algorithm to calculate the gradient values in the *x* and *y* directions, and the 3 * 3 matrix window is used to screen the candidate corners twice to extract the small boundaries in the image. Finally, the final real corners are determined by nonmaximum suppression.

The specific arrangement of the paper is as follows: Section 2 introduces the FAST algorithm and the improved FAST algorithm. In Section 3, the Harris algorithm and the improved Harris algorithm are introduced. In Section 4, the mathematical model of corner detection combined with FAST and Harris is introduced in detail. In Section 5, simulation experiments are carried out on the PyCharm platform, and the detection error rate, positioning error, repeatability, and execution time performance of 9 corner detection methods are compared. Finally, the conclusion is given in Section 6.

2. Improved FAST Algorithm for Rough Capture

2.1. FAST Algorithm Mathematical Model. The classical FAST algorithm [33] is shown in Figure 1, and Bresenham circle is drawn by considering the target detection point $P(x, y)$ as the center of a circle and three pixels as radius. The corners are obtained by judging the difference between the pixel value $I_p(x_p, y_p)$ of the $P(x, y)$ point and the pixel value $I_i(x_i, y_i)$ of the discrete points P_1~P_{16} on the Bresenham circle. A large part of the corners detected by the FAST algorithm is error corners, or redundant corners.

Hypothesis 1. The pixel values of P_1~P_{16} are $I_i(x_i, y_i)$ ($i = 1$, 2, 3 ... 16).

$$\left|I_i(x_i, y_i) - I_p\left(x_p, y_p\right)\right| \geq T. \tag{1}$$

$$N = \sum_{x\forall(\text{circle}(p))} \left|\left|I_i(x_i, y_i) - I_p\left(x_p, y_p\right)\right| > T\right| \geq 12, \tag{2}$$

where N is the number of pixels satisfying (1), T is the threshold, and the empirical value is usually 50 pixels, where $N > 12$, 12 is the best experimental result value obtained through experimental comparison in literature [33].

When the pixel value of point $P(x, y)$ satisfies (2), then $P(x, y)$ is judged to be a corner point.

The corner detection rate of the classic FAST algorithm is very high, but the corner response function of the FAST algorithm is shown in (2). When N meets the condition, there are more corners, and many feature points are connected together, resulting in redundant corners. More than that, it seriously affects the detection performance of FAST.

2.2. Mathematical Model of Rough Capture FAST Algorithm. The algorithm proposed in the paper reduced the number of points compared with the target detection point $P(x, y)$ in (1) to improve the corner detection speed of FAST algorithm; the number of the contrast points between the center point and on the circumference discrete point is decreased by 25%. As shown in Figure 1, only the difference between the point $P_1(x_1, y_{1-3})$, $P_5(x_{5+3}, y_5)$, $P_9(x_9, y_{9+3})$ and $P_{13}(x_{13-3}, y_{13})$ on the Bresenham circle and $P(x, y)$ should be estimated, which

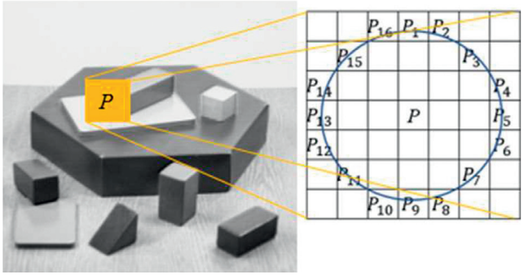


Figure 1: Selection of FAST judgment points.

is equivalent to the vertical point of the 7 * 7 matrix window judged with P (x, y) as the center. Equation (3) is the FAST corner response function. P (x, y) is the candidate corner if the value of the pixel is satisfied.

$$N = \sum_{x\forall(\text{circle}(p))} \left| \left| I_i(x_i, y_i) - I_p\left(x_p, y_p\right) \right| > T \right| \geq 3, \tag{3}$$

where I_i $(i = 1{,}5{,}9{,}13)$ represents the pixel value of any pixel on the circumference. I_{p} is the pixel value of the center point P (x, y). Set the threshold $T = 50$. Roughly capturing FAST algorithm reduces 12 P (x, y) contrast points compared with the classical FAST algorithm and speeds up the detection speed, but it gets more candidate corners and the selection of candidate corners is not optimal. The improved Harris algorithm is adopted to get the accurate corner points in the paper and finally used SAD for nonmaximum suppression to get the optimal corner point.

2.3. Target Image Padding. As proposed in [36], the candidate corners are captured roughly by the FAST algorithm preliminarily as shown in Figure 2(a), the 7 * 7 matrix window is selected to slide the target image, and three rows and three columns of pixels are lost in each edge of the target image. In Figure 2, the matrix box in blue is the target image, while the matrix box with the dotted line in red is the padded target image.

It is necessary to pad the image for obtaining each point P (x, y) in the image as shown in Figure 2(b); the image padding size should be expressed in (4) if the input and output sizes are the same.

$$n = \lfloor \frac{n + 2q - f}{s} \rfloor + 1. \tag{4}$$

In (4), n * n is the input size of the image; f is the size of the matrix window; q is the size of padding; s is step length; $\lfloor\rfloor$ is round down to the integer.

Hypothesis 2. Image size is *rows* * *cols*, padding the image in horizontal direction with the parameters $n = \mathit{rows}, f = 7, s = 1$, and putting the parameters into (4). Then, we get

$$\text{rows} = \text{rows} + 2q - 7 + 1 \longrightarrow q = 3. \tag{5}$$

The rows in (5) are replaced by clos and the number of padded columns is shown in (3). According to (5), the size of the image padding value is related to the window size of the sliding matrix. Therefore, the edge of the target image is padded with three rows and three columns to ensure each pixel in the image is detected, as shown in Figure 2(c).

Figure 3 shows the process diagram of rough capture FAST algorithm.

Step 1. Padding the original image as shown in Figure 3(a).

Step 2. As shown in Figure 3(b), move the matrix window of 7 * 7 from left to right, top to bottom, and step distance is one pixel, according to (3) to determine corner point.

Step 3. Get the candidate corner position as shown in Figure 3(c).

The padding methods of roughly capturing FAST include constant padding, symmetry axis padding, and boundary pixel value padding. For the same corner points, the running time of the constant padding method is superior to the other two methods by 33.31% and 3.59% according to the simulation, so the constant padding method is adopted in the paper.

Figure 4 shows the format of the pseudocode for Algorithm 1 rough capture FAST algorithm.

3. Improved Harris Algorithm for Accurate Capture

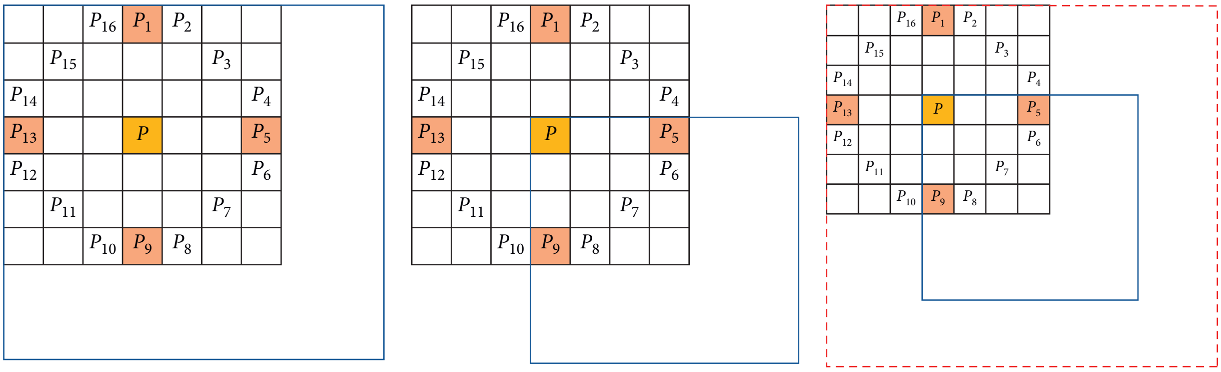
The above rough capture FAST algorithm is used to get the candidate corner set after padding the target image with constant. The candidate corner set includes a lot of redundancy corners and there are some corner points palletized, so it is necessary to further screen the corners. The classical Harris algorithm has good robustness and using Scharr operator [40] instead of Sobel operator can screen the real corners more carefully in the candidate corner set.

3.1. Mathematical Model of Harris Algorithm. The image is observed by sliding a small window in classic Harris algorithm [18]. The window moves to arbitrary directions, and the points whose gray changed obviously should be the corners. The pixel gray of P (x, y) changed with the moving of the window based on Taylor series expression showed in

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2, \tag{6}$$

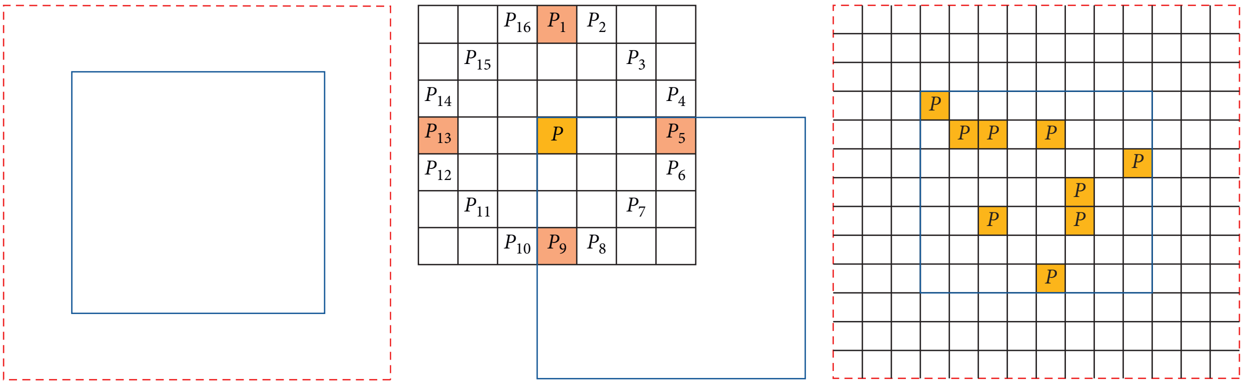
where u and v are the offsets in vertical and horizontal directions; w (x, y) represents the weight of point-centered; I represent the gray value.

Equation (6) is expanded by a first-order Taylor expansion and can be rewritten as (7) approximately if the image I (x, y) translated Δx in a horizontal direction while Δy in vertical direction.



(a) (b) (c)

Figure 2: Target image padding: (a) Bresenham circle position, (b) expected position, and (c) postpadding position.



(a) (b) (c)

Figure 3: Improved FAST corner detection flow chart: (a) image padding, (b) corner point judgment, and (c) corner point position.

```
Algorithm 1: Rough capture FAST algorithm
Input: image, $T$=50
Output: PointList
  1.The target image edge padded with 3 pixels and the padding constant is 0.
  2.rows, cols = image. shape [:2]
  3.Consider the target pixel as the circle center and 3 pixels as the radius, compared the four pixels
  numbered 1, 5, 9 and 13 with the target pixel. col, row = point.
  4.for row in range (rows):
  5.for col in range (cols):
  6.if $N = \sum_{x\forall (circle\,(\text{point}))} ||I_i\,(row_i, col_i) - I_p\,(row_p, col_p)|{>}\text{T}| \geq 3$ :
  7.        PointList.append((col, row))
  8. return PointList;
```

Figure 4: Pseudocode format of Algorithm 1.

$$E(u,v) \approx \sum_{x,y} w(x,y)\left[I(x,y) + \frac{\partial I}{\partial x}(x,y)u + \frac{\partial I}{\partial y}(x,y)v - I(x,y)\right]^2. \quad (7)$$

$$\approx \sum_{x,y} w(x,y)\left[\frac{\partial I}{\partial x}(x,y)u + \frac{\partial I}{\partial y}(x,y)v\right]^2. \quad (8)$$

Equation (8) is obtained by second-order Taylor expansion.

$$\begin{aligned} E(u,v) &= \sum_{x,y} w(x,y)\left(u^2 f_x^2(x,y) + 2\,\mathrm{uvf}_x(x,y) f_y(x,y) + v^2 f_x^2(x,y)\right) \\ &= \sum_{x,y} w(x,y)\left(u^2 I_x^2 + 2\,\mathrm{uvI}_x I_y + v^2 I_y^2\right) \\ &= \begin{bmatrix} u & v \end{bmatrix}\left(\sum_{x,y} w(x,y)\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} H \begin{bmatrix} u \\ v \end{bmatrix}. \end{aligned} \quad (9)$$

In (9) H is composed of a window function and a horizontal vertical gradient.

$$H = \sum_{x,y} w(x,y)\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \longrightarrow A^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}A. \quad (10)$$

Equation (10) is the diagonal real symmetric matrix, A is the rotation factor, and the variation components in two orthogonal directions namely λ_1 and λ_2 (eigenvalue) are extracted.

In (11) R represents the Harris corner response function, and α is a constant (generally 0.04–0.06), set as $\alpha = 0.05$ in the paper.

$$R = \det(H) - \alpha(\mathrm{trace}(H))^2, \quad (11)$$

where $\det(H) = \lambda_1\lambda_2$ and $\mathrm{trace}(H) = \lambda_1 + \lambda_2$.

3.2. Mathematical Model of Accurate Capture Harris Algorithm. The Scharr operator is used to calculate the gradient values of the H matrix in the x and y directions instead of Sobel operator to improve the detection accuracy of Harris algorithm. The Scharr operator can extract the tiny boundaries effectively and then filter the H matrix with Gaussian filter to eliminate the isolated points and convex points in the image.

Gaussian filtering function is used as the window function w (x, y) in (10) expressed in

$$w(x,y) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}. \quad (12)$$

Table 1 is the parameters of the Gaussian filter. The template on the left is a 3* * *3 matrix, which is the window function calculated by (12) assuming the value of σ is 1.5. And then the template on the right side is obtained after normalizing the left one, which is used to eliminate isolated points and raised points in candidate corner points with the normalized filter.

E (u, v) is related to I_x and I_y, and the H function in (10) can be simplified as

$$H = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (13)$$

In (13), I_x uses the 3 * 3 Scharr gradient operator [−3 0 3; −10 0 10; −3 0 3] [40] to calculate the horizontal gradient, and I_y uses the 3 * 3 Scharr gradient operator [−3 −10 −3; 0 0 0; 3 10 3] [40]. Calculate the vertical gradient.

In (14), R is the Harris corner response function, and α is a constant; α is 0.05 in the paper.

$$R = \det(H_0) - \alpha(\mathrm{trace}(H_0))^2, \quad (14)$$

where H_0 is the new matrix obtained with transformation as in Table 1 Gaussian filtering.

Finally, whether P (x, y) is a real corner point or not can be judged according to (18).

$$\begin{cases} R \text{ is much greater than } 0, \text{corner}, \\ R \text{ is much less than } 0, \text{edge}, \\ |R| \approx 0, \text{plane}. \end{cases} \quad (15)$$

The flow chart of accurate Harris corner detection algorithm is shown. In Figure 5, Steps 1–4 keep on cycling until the candidate corner set is empty:

(1) Step 1: take one of the candidate corners P (x, y) obtained by roughly capturing FAST algorithm

(2) Step 2: calculate x-axis and y-axis gradient values of P (x, y) points using 3 * 3 Scharr operator

(3) Step 3: a new H_0 matrix is obtained by Gaussian filtering of x, y gradient values

Table 1: Parameters of the Gaussian filter.

(−1, −1)	(−1, 0)	(−1, −1)		0.0585	0.0965	0.0585
(0, −1)	(0, 0)	(0, 1)	$\xrightarrow{w(x,y)}$	0.0965	0.1592	0.0585
(1, −1)	(1, 0)	(1, 1)		0.0585	0.0965	0.0585

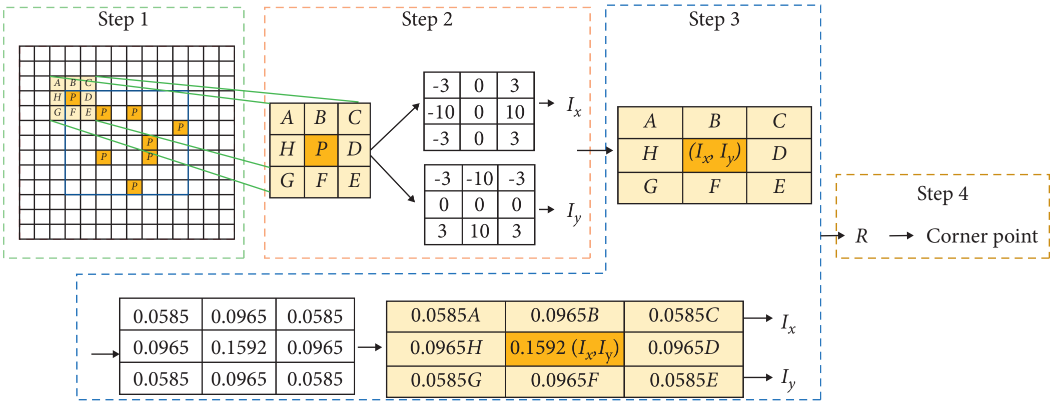


Figure 5: Flow chart of improved Harris algorithm corner detection.

(4) Step 4: according to corner criterion R, determine whether point P (x, y) is a corner or not

Figure 6 shows the format of the pseudocode for Algorithm 2 accurate capture Harris algorithm.

4. FAST-Harris Fusion Corner Detection Algorithm

The corner detection algorithm fusion combining FAST with Harris is proposed in the paper. According to the improved FAST algorithm in Section 2.2, the corner detection is captured by the matrix window of 7 * 7 to obtain the candidate corner set. Then the improved Harris algorithm in Section 3.2 is used to rescreen the candidate corner by the matrix window of 3 * 3. Finally, the real corner is determined by nonmaximum suppression.

4.1. FAST-Harris Algorithm Mathematical Model. The mathematical model of FAST-Harris algorithm is divided into two parts. In the first part, the candidate corner set is obtained if the pixels in the image are satisfied (16). In the second part, if the candidate corner points are satisfied (16), then screen the point according to (17), and finally the real corners are obtained.

$$\text{Num} = \sum_{x\forall(\text{circle}(p))} \left| \left| I_i(x_i, y_i) - I_p\left(x_p + q, y_p + q\right) \right| > T \right| \geq 3 \longrightarrow P\left(x_p + q, y_p + q\right). \tag{16}$$

$$R_{P\left(x_p+q, y_p+q\right)} = \det\left(H_{P\left(x_p+q, y_p+q\right)}\right) - \alpha\left(\text{trace}\left(H_{P\left(x_p+q, y_p+q\right)}\right)\right)^2 > 0 \longrightarrow P_c\left(x_p + q, y_p + q\right). \tag{17}$$

$$H_{P\left(x_p+q, y_p+q\right)} = \begin{bmatrix} I^2_{x_p+q} & I_{x_p+q} I_{y_p+q} \\ I_{x_p+q} I_{y_p+q} & I^2_{y_p+q} \end{bmatrix}. \tag{18}$$

In (16) and (17), I_i $(i = 1, 5, 9, 13)$ denotes the pixel value of the pixel on the circumference with radius 3 pixels; I_p is the pixel value of the center point P $(x_p + q, y_p + q)$, q is the padding size for the target image; T is the threshold. $P(x_p + q, y_p + q)$ is the candidate corner; $P_c(x_p + q, y_p + q)$ is the corner after the Gaussian filter; I_{x_p+q} and I_{y_p+q} are the horizontal and vertical one-step pixel value of $P(x_p + q, y_p + q)$.

4.2. Nonmaximum Suppression. The corner set is obtained by the corner detection algorithm based on FAST-Harris fusion above. In order to reduce the redundancy corners, SAD is used for nonmaximum suppression.

Calculate the SAD values of each candidate corner $P_c(x_p + q, y_p + q)$ and 16 pixels value around its Bresenham circle as the scores of the candidate corner, expressed as in (18).

$$S(P, A)_{\text{SAD}} = \sum_{i=1}^{16} \left| I_P\left(x_P + q, y_P + q\right) - I_i\left(x_i, y_i\right) \right|. \tag{19}$$

```
Algorithm 2: Accurate capture Harris algorithm
Input: PointList, $\alpha$=0.05
Output corners
  1. Using 3*3 Scharr gradient operator to calculate row, coldirection gradient value. For the candidate
     corner saved in PointList [].
  2. Find the values of $I^2_{row_p}$, $I^2_{col_p}$, $I_{row_p}$ and $I_{col_p}$ in $H_{P(row_p,\, col_p)}$ matrix.
  3. Gaussian filtering is adopted to eliminate isolated points and bulges and build up a new matrix $H_0$.
  4. Calculate the corner response function $R_P\ (row,col) = det\ (H_{P(row,col)})\text{-}\alpha\ (trace\ (H_{P(row,col)}))^2$
  5. $\alpha$=0.05
  6. if $R_{P(row,col)}$ >> 0: return corners;
       else $R_{P(row,col)}$ << 0: return edge;
       else $|RP(row,col)| \approx 0$: return plane;
  7.Perform SAD non-maximum suppression on corners= [] filtered the redundant corners.
```

Figure 6: Pseudocode format of Algorithm 2.

In (19), I_i is the pixel value of discrete point on the Bresenham circle; I_p is the pixel value of candidate corner.

According to the Euler distance expressed as (20), the two candidate corners $P_c\ (x_i, y_i)$ and $P_c\ (x_j, y_j)$ are adjacent if the Euler distance between $P_c\ (x_i, y_i)$ and $P_c\ (x_j, y_j)$ is less than four pixels.

$$e = \sqrt{\left(x_i - x_j\right)^2 - \left(y_i - y_j\right)^2} < 4. \tag{20}$$

Finally, comparing the scores of two adjacent corners, the corner with low score should be desert. Using the nonmaximum suppression to find the local maximum value, reduce the redundant corners and then determine the final real corners.

4.3. Steps of FAST-Harris Algorithm Corner Detection. The flow chart of corner detection algorithm proposed in this paper is shown in Figure 7. First, the edge of the target image is padded, then the FAST algorithm (green process box) is roughly captured, and then the Harris algorithm (yellow process box) is accurately captured. Finally, the nonmaximum suppression (gray process box) is used to obtain the final corner point.

5. Analysis of Experimental Results

This paper selects PyCharm-python3.8 as the experimental development platform. Because the stability of the initial program is not high, all the experimental results in this paper are carried out after the program runs 20 times. The algorithm parameters are taken as the empirical value $T = 50$ pixels, $q = 3$. The target image padding method selects constant padding, and all parameters in the experiment are consistent. The proposed method was compared with the other eight corner detectors (Harris [18], FAST [33], ORB [34], SURF [30], SIFT [29], Shi-Tomasi [26], Subpixel level [32], and AKAZE [31]). The method in the paper is compared with Harris [18], Shi-Tomasi [26], and Subpixel level [32] represented by improving positioning accuracy; SIFT [29] represented by improving robustness, SURF [30] and AKAZE [31] represented for comparison; FAST [33] algorithm represented by detection speed and ORB [34] for comparison. After 50 times running for the program, the performance evaluation for error detection rate, positioning error, robustness, and running time is obtained.

5.1. Error Detection Rate and Positioning Error Performance Evaluation of Corner Detection. In this section, the error detection rate and localization error are used to evaluate the nine methods (Harris [18], FAST [33], ORB [34], SURF [30], SIFT [29], Shi-Tomasi [26], Sub -pixel level [32], AKAZE [31], and F–H).

5.1.1. Error Detection Rate. In the image corner detection results, the ratio of error corner F to the total number of detected corners is represented by Er. If the total number of corner points got by corner detection algorithm is S, the number of correct matching corner points is R, the number of error corner points is F (including the number of missing and error corner points), and the number of real corner points detected manually is $\widehat{R}$. Er can be expressed as

$$\mathrm{Er} = \frac{F}{S} \times 100\%. \tag{21}$$

Equation (21) matches the $\widehat{R}$ corner point set with the position coordinates of the S corner point set, keeping the real corner detected within 4 pixels in the R corner set, with distance over 4 pixels, less than 8 pixels for missed corner; otherwise, it is error corer.

Figure 8 shows three test images: ’Block image’ [41], ’Checkerboard image’ [42], ’House image’ [41], and corresponding manual marked corners (marked with green circles). The test images ’Block image’, ’Checkerboard image’, and ’House image’ contain 59, 100, and 61 corners, respectively. The manually marked corner position is jointly marked by 10 experts. When eight experts mark the same corner position, it is the real corner position [16].

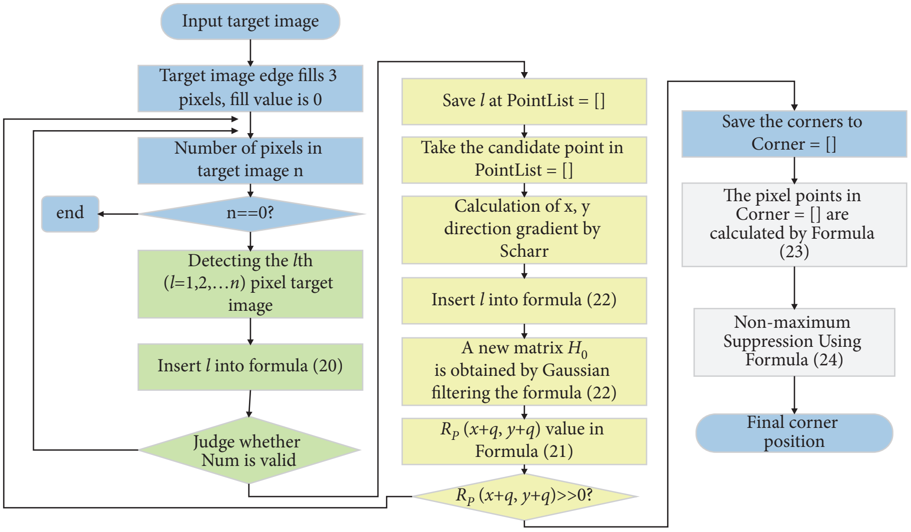
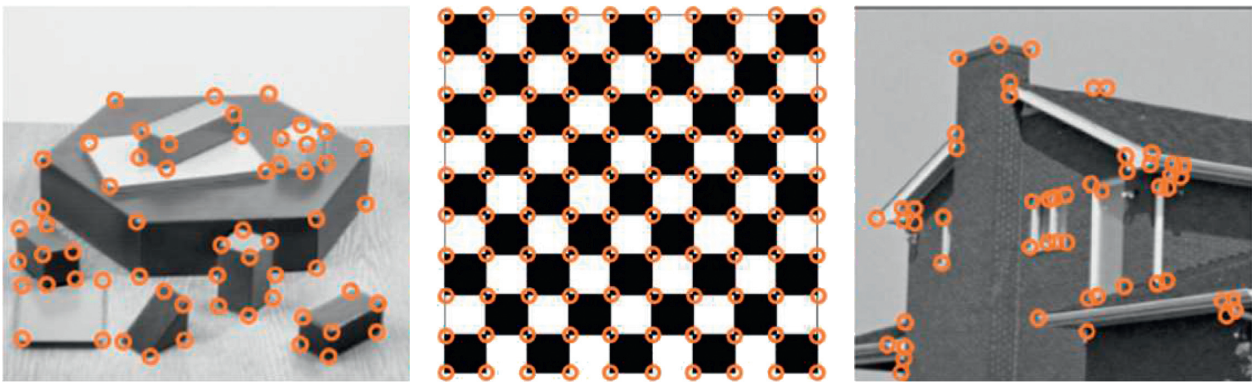

Input target image
Target image edge fills 3 pixels, fill value is 0
Number of pixels in target image n
end
n==0?
Detecting the *l*th (*l*=1,2,…*n*) pixel target image
Insert *l* into formula (20)
Judge whether Num is valid
Save *l* at PointList = []
Take the candidate point in PointList = []
Calculation of x, y direction gradient by Scharr
Insert *l* into formula (22)
A new matrix H_0 is obtained by Gaussian filtering the formula (22)
R_p (*x*+*q*, *y*+*q*) value in Formula (21)
R_p (*x*+*q*, *y*+*q*)>>0?
Save the corners to Corner = []
The pixel points in Corner = [] are calculated by Formula (23)
Non-maximum Suppression Using Formula (24)
Final corner position

Figure 7: FAST-Harris algorithm flow chart.



(a) (b) (c)

Figure 8: Real corner position of three test images: (a) block image, (b) checkerboard image, and (c) house image.

The corner detection results of the algorithm in this paper and other 8 algorithms on the three target images are shown in Figures 9–11. As shown in Table 2, Harris [18], Shi-Tomasi [26], Subpixel level [32], SIFT [29], SURF [30], AKAZE [31], FAST [33], ORB [34], F–H, and the number of error detection corners on the 'Block image' [41] test images of the proposed algorithm are 29, 25, 28, 99, 42, 63, 46, 38, and 4, respectively. The number of misdetection points on the 'Checkerboard image' [42] test images is 12, 12, 10, 276, 37, 130, 107, 48, and 4, respectively. The number of misdetection points on the 'House image' [41] test images is 45, 32, 30, 56, 36, 52, 51, 36, and 25, respectively.

As shown in Figure 12, compared with the Harris [18] algorithm, Shi-Tomasi [26] and Subpixel level [32] algorithms are represented by the positioning accuracy, and the average error detection rate is reduced by 16.89%, 33.08%, and 16.87%. Compared with the other five algorithms, the algorithm in this paper has the best error detection rate, and the average error detection rate is 24.60%.

5.1.2. Positioning Error. The positioning error [43] is an important index for evaluating the accuracy of corner detection algorithm. Assuming that the $\widehat{R}$ corner set $\{(\widehat{x}_i, \widehat{y}_i), i = 1,2, \ldots, m\}$ and the S corner set $\{(x_i, y_i), i = 1,2, \ldots, m\}$ are matched successfully, the localization error can be defined as

$$E = \sqrt{\frac{1}{m} \times \sum_{i=1}^{m} \left[\left(\widehat{x}_i - x_i\right)^2 + \left(\widehat{y}_i - y_i\right)^2 \right]}. \quad (22)$$

As shown in Figure 13, the positioning error of the F–H algorithm is the lowest on the three images. As shown in Table 3, the average positioning errors of the 9 algorithms in the three images are 2.45, 1.92, 1.94, 2.72, 2.63, 1.91, 2.01, 1.93, and 1.42 pixels, respectively. Compared with the Harris [18] Shi-Tomasi [26], and the Subpixel level [32] represented by the positioning accuracy, the average positioning error is reduced by 42.04%, 26.04%, and 26.80%, respectively. Compared with the other five algorithms, the algorithm in this paper has the smallest average positioning error.

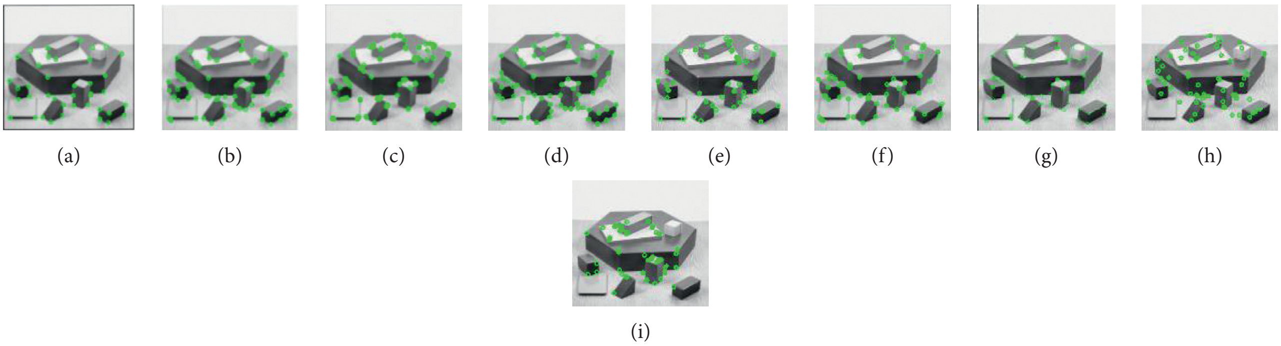

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 9: “Block image” test results of 9 algorithms: (a) F–H, (b) Harris, (c) FAST, (d) Shi-Tomasi, (e) SURF, (f) Subpixel level, (g) ORB, (h) SIFT, and (i) AKAZE.

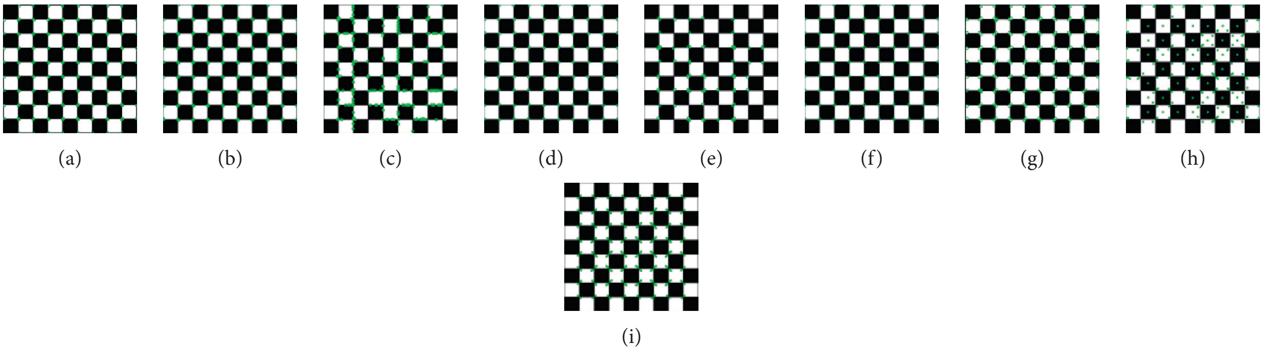

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 10: “Checkerboard image” test results of 9 algorithms: (a) F–H, (b) Harris, (c) FAST, (d) Shi-Tomasi, (e) SURF, (f) Subpixel level, (g) ORB, (h) SIFT, and (i) AKAZE.

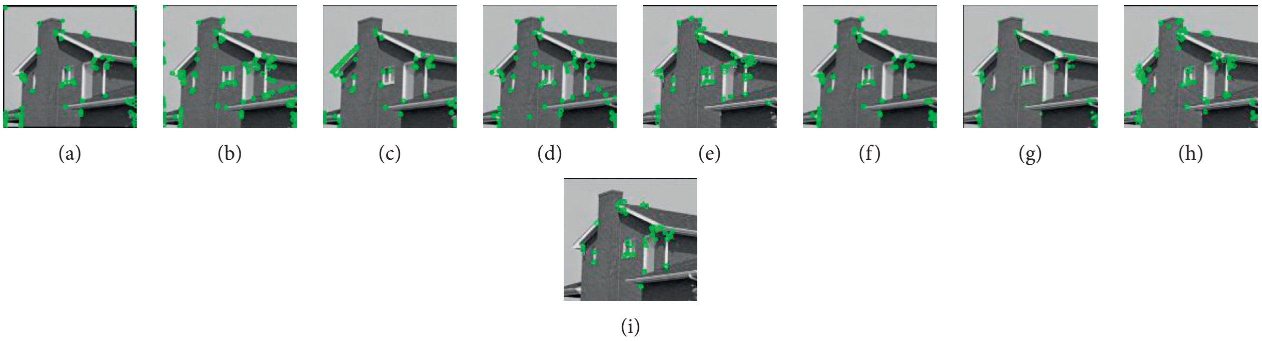

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 11: “House image” test results of 9 algorithms: (a) F–H, (b) Harris, (c) FAST, (d) Shi-Tomasi, (e) SURF, (f) Subpixel level, (g) ORB, (h) SIFT, and (i) AKAZE.

5.2. Robustness Evaluation under Image Transformation. In this section, repeatability is used to evaluate the robustness of the 9 methods (Harris [18], Shi-Tomasi [26], Subpixel level [32], SIFT [29], SURF [30], AKAZE [31], FAST [33], ORB [34], and F–H). The repeatability under image transformation in [43] means that the image can also be detected through rotation, scaling, noise, and lossy JPEG compression. This is a robust corner detection method. Let n_1, n_2 be the number of corner points detected in the two pictures and R the number of corner points that appear in the two pictures at the same time, and (23) is the repeatability under image transformation.

$$\text{REPEAT} = \frac{1}{2} \times R \times \left(\frac{1}{n_1} + \frac{1}{n_2} \right). \tag{23}$$

Due to the different corner detection methods, the accuracy of the corner position is different, so as long as the corner appears within 4 pixels adjacent to the target position, it is considered that the corner appears repeatedly. The higher the repeatability under image transformation, the better the robustness of the detection method.

As shown in Figure 8, robustness experiments are performed on “Block image” [41], “Checkerboard image” [42], “House image” [41] images. Each image undergoes 4

Table 2: Comparison of error detection rates (%) of 9 detection algorithms on three test images.

Method	Missed corners	Error corners	Error detection rate (%)
		Block image	
Harris [18]	13	16	42.65
Shi-Tomasi [26]	**10**	15	38.46
Subpixel level [32]	14	14	40.00
SIFT [29]	43	56	76.71
SURF [30]	27	15	82.35
AKAZE [31]	33	30	43.10
FAST [33]	17	29	37.70
ORB [34]	13	25	36.89
F–H	15	**0**	**36.59**
		Checkerboard image	
Harris [18]	12	**0**	7.69
Shi-Tomasi [26]	12	**0**	13.64
Subpixel level [32]	10	**0**	5.95
SIFT [29]	32	244	58.09
SURF [30]	37	**0**	17.96
AKAZE [31]	35	95	30.45
FAST [33]	32	75	39.34
ORB [34]	28	20	25.32
F–H	**4**	**0**	**2.50**
		House image	
Harris [18]	**7**	38	38.46
Shi-Tomasi [26]	20	12	58.18
Subpixel level [32]	23	**7**	42.85
SIFT [29]	26	30	30.30
SURF [30]	23	13	64.28
AKAZE [31]	20	32	60.37
FAST [33]	21	30	61.45
ORB [34]	12	24	47.25
F–H	**7**	18	**34.72**

Bold shows minimum missed corners, minimum error corners, and minimum error detection rate (%).

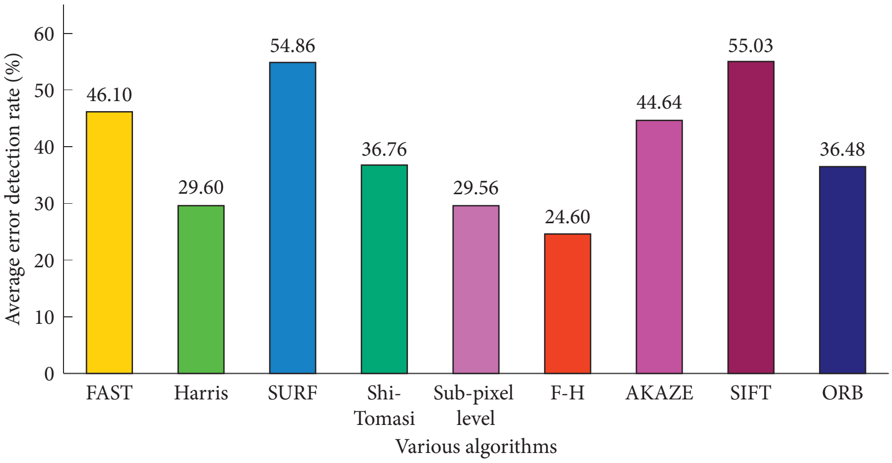


Figure 12: Average error detection rate (%) comparison of 9 detection algorithms on three test images.

different transformations, as shown in Figure 14, and a total of 108 test images are obtained.

Rotation angle: the original image in the $[-\pi, \pi]$ range, $\pi/4$ angle to rotate the test image.

Uniform scale factor: the horizontal and vertical resolution of the image, between $[-1, 1]$ with 0.2 interval test image.

Add Gaussian noise: add Gaussian noise to simulate sensor noise caused by bad lighting and high temperature, add zero-mean Gaussian noise in the range of [0, 3.5], and get test images at intervals of standard deviation 0.5.

JEPG quality factor: the 'cv2.IMWITE_JPEG_ QUALITY, C' function is used to ensure that the image size is fixed.

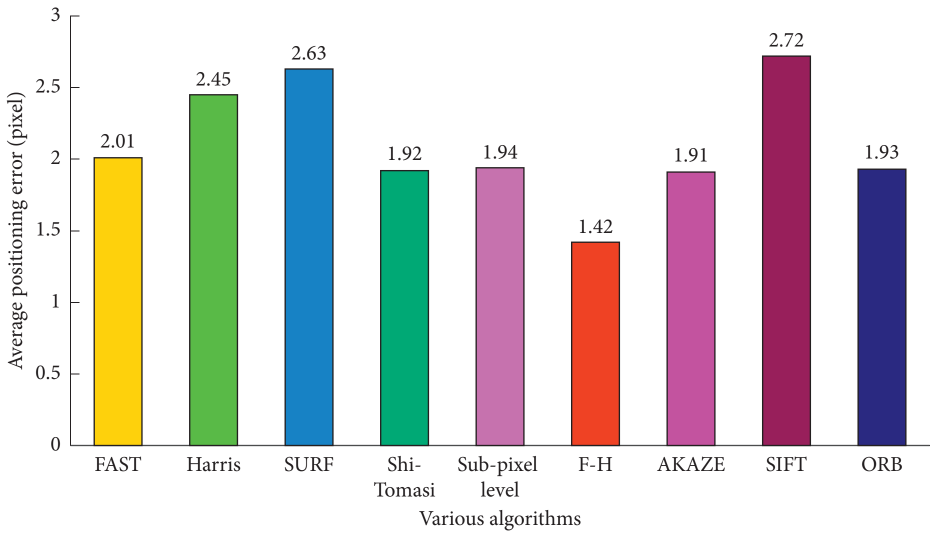


Figure 13: Average localization error (pixels) comparison of 9 detection algorithms on three test images.

Table 3: Comparison of positioning errors (pixels) of 9 detection algorithms on three test images.

	Method								
	Harris and Stephens [18]	Shi and Tomasi [26]	Subpixel level [32]	SIFT [29]	SURF [30]	AKAZE [31]	FAST [33]	ORB [34]	F–H
Block image	2.51	1.97	1.59	2.47	2.34	1.99	2.28	2.10	**1.33**
Checkerboard image	2.23	1.68	2.09	2.99	2.96	1.79	1.60	1.59	**1.54**
House image	2.62	2.10	2.14	2.72	2.60	1.97	2.15	2.07	**1.38**

Bold shows minimum positioning error (pixels).

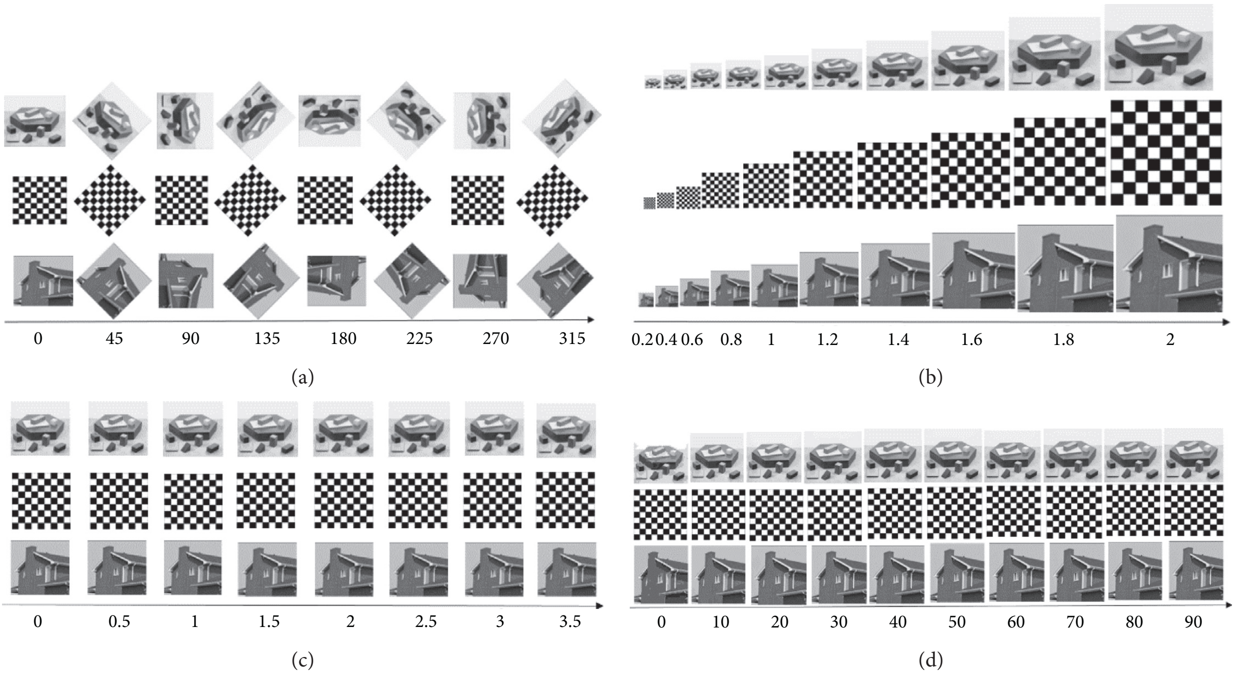


Figure 14: Image sets under different transformations: (a) rotation angle, (b) uniform scale factor, (c) Gaussian noise, and (d) JEPG quality factor.

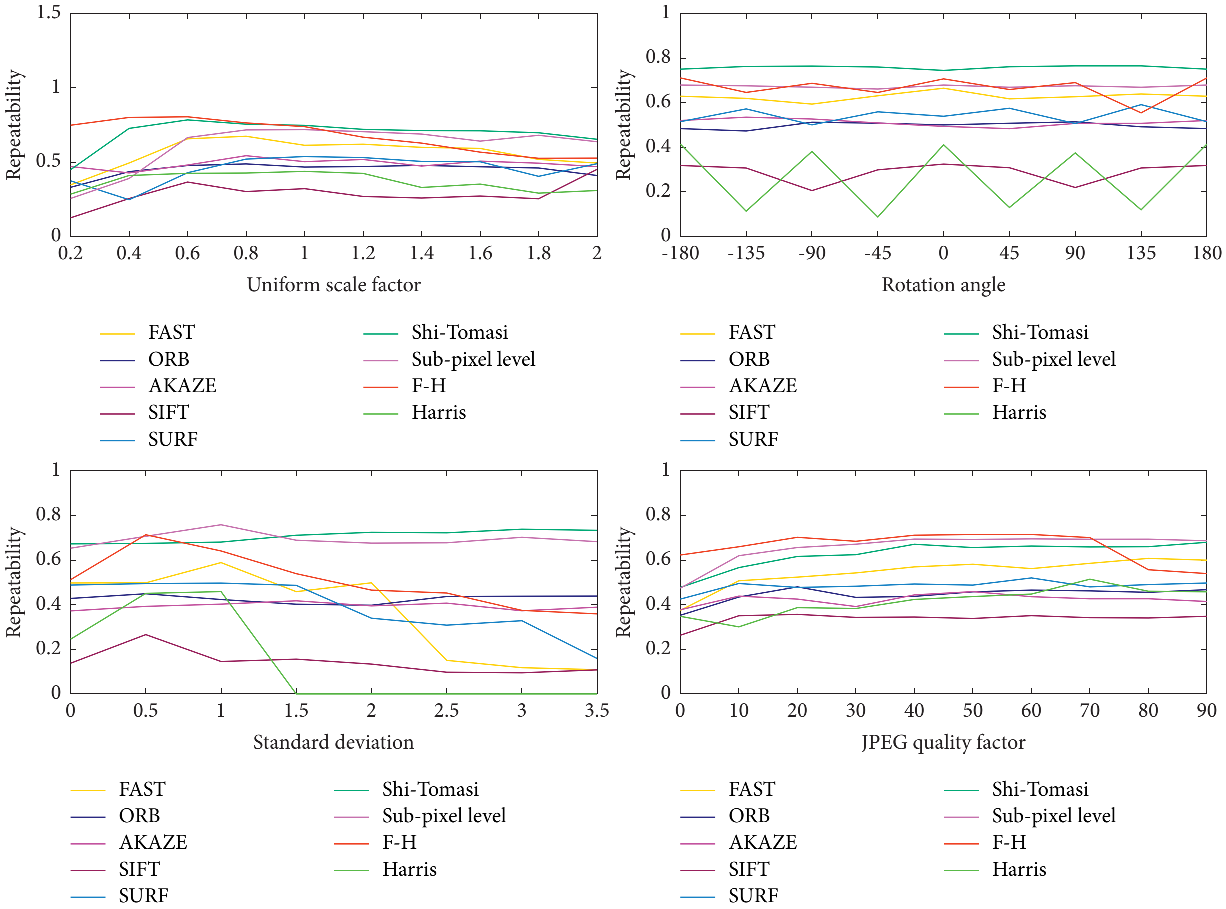


Figure 15: Line chart of robustness for 9 detection algorithms with 3 groups of images.

Table 4: The robustness of three groups of images under different transformations.

Method	Uniform scale factor	Rotation angle	Add Gaussian noise	JEPG quality factor
Harris [18]	0.37	0.25	0.14	0.42
Shi-Tomasi [26]	**0.69**	**0.76**	**0.71**	0.63
Subpixel level [32]	0.61	0.67	0.69	0.65
SIFT [29]	0.31	0.29	0.14	0.34
SURF [30]	0.45	0.55	0.39	0.49
AKAZE [31]	0.48	0.51	0.39	0.42
FAST [33]	0.56	0.63	0.36	0.55
ORB [34]	0.44	0.50	0.42	0.44
F–H	0.68	0.68	0.51	**0.66**

Bold shows the optimal robust values of three groups of images under different transformations.

The image quality C changes within the range of [0, 90], and the test image is obtained at the interval of 10.

Figure 15 shows the line chart of robustness experimental results of 9 detection algorithms for “Block image” [41], “Checkerboard image” [42], and “House image” [41] images. Table 4 illustrates the data of the experimental results quantized value; it can be seen that the algorithm in this paper is compared with the SIFT [29] algorithm, SURF [30] and AKAZE [31] algorithm represented by improving robustness. Robustness is optimal. Among the 9 algorithms, the algorithm proposed in this paper has the second performance in uniform scaling and rotation, the third performance in adding Gaussian noise, but the best performance in lossy JPEG compression.

To further verify the robustness of the algorithm, as shown in Figure 16, 10 images of different scenes were selected in the COCO2017 dataset [44] to verify the average repeatability of the corner detection algorithm. Each image was subjected to 4 different transformations, resulting in a total of 360 test images.

Figure 17 shows a line graph of the robustness of 9 detection algorithms for 10 groups of images, and Table 5 shows the data quantification values. From Table 5, it can

Figure 16: COCO2017 test data set (Reference [44]).

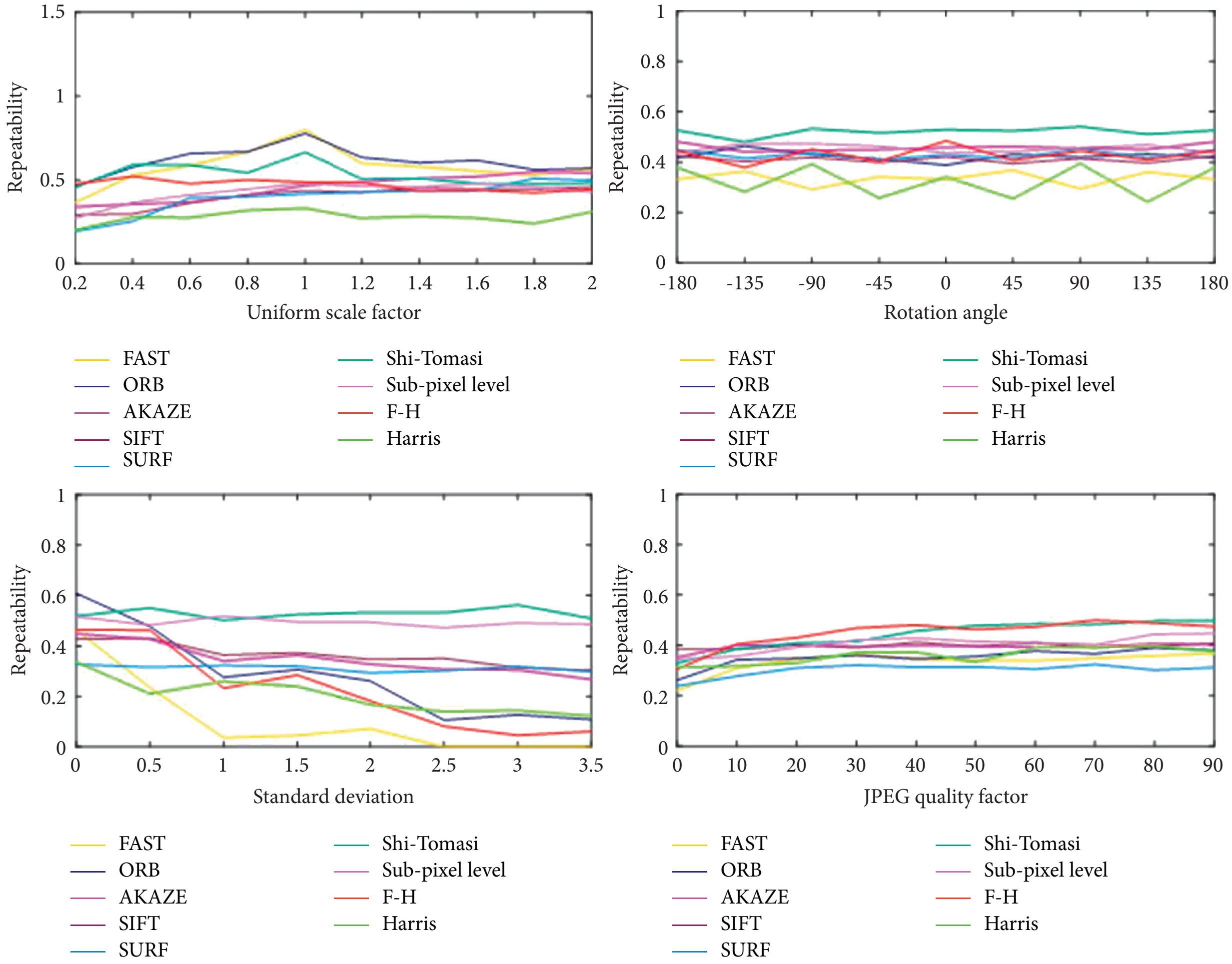


Figure 17: Line chart of robustness of 9 detection algorithms for 10 groups of images.

be seen that the algorithm in the paper compared with the SIFT [29], SURF [30], and AKAZE [31] is represented by improved robustness. The robustness of the proposed algorithm is the best in uniform scaling and lossy JPEG compression. Among the 9 algorithms, the proposed algorithm is the best in lossy JPEG compression.

5.3. Performance Evaluation of Corner Detection Running Time.

This section uses runtime to evaluate the performance of nine methods (Harris [18], Shi-Tomasi [26], Subpixel level [32], SIFT [29], SURF [30], AKAZE [31], FAST [33], ORB [34], and F–H). As shown in Figure 18, eight different scenes of “Construction”, “Train”, “Wall”, “Airplane”, “Fruit”,

Table 5: The robustness of 10 groups of images under different transformations.

Method	Uniform scale factor	Rotation angle	Add Gaussian noise	JEPG quality factor
Harris [18]	0.28	0.32	0.20	0.36
Shi-Tomasi [26]	0.53	**0.52**	**0.53**	0.44
Subpixel level [32]	0.43	0.45	0.49	0.41
SIFT [29]	0.40	0.41	0.36	0.40
SURF [30]	0.40	0.43	0.31	0.30
AKAZE [31]	0.46	0.46	0.35	0.39
FAST [33]	0.57	0.34	0.12	0.33
ORB [34]	**0.61**	0.42	0.28	0.35
F–H	0.47	0.43	0.23	**0.45**

The best robust values of 10 groups of images under the transformation of uniform scale factor, rotation angle, add Gaussian noise, and JEPG quality factor (the higher the value, the better the robustness).

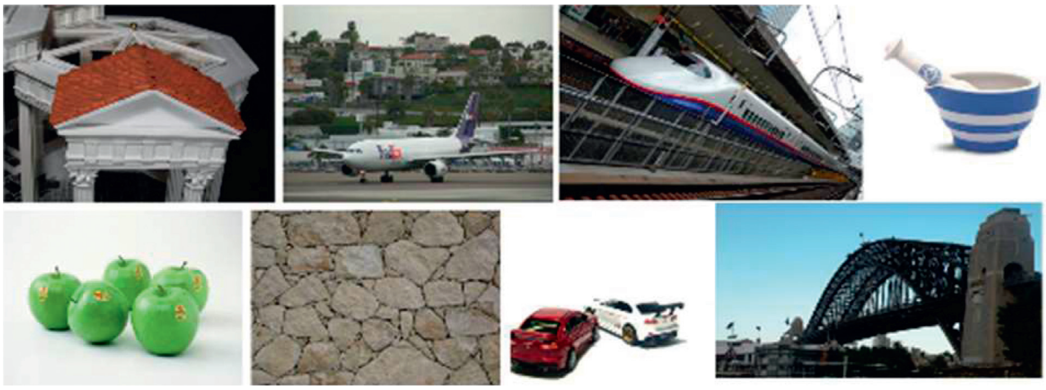


Figure 18: ImageNet test dataset (Reference [45]).

Table 6: Performance comparison of nine detection methods in running time (s).

Method	Image											
	Construction	Airplane	Train	Instrument	Bridge	Fruit	Wall	Car	Block	Checkerboard	House	Average value
Harris [18]	23.17	23.23	26.90	37.68	23.15	44.94	23.12	27.62	10.89	10.72	10.93	23.85
Shi-Tomasi [26]	0.07	0.07	0.05	0.07	0.07	0.08	0.06	0.06	0.03	0.03	0.02	0.06
Subpixel level [32]	0.07	0.06	0.04	0.02	0.03	0.03	0.03	0.02	**0.01**	**0.01**	**0.01**	0.03
SIFT [29]	0.13	0.18	0.10	0.13	0.11	0.14	0.10	0.10	0.04	0.03	0.04	0.10
SURF [30]	0.08	0.09	0.07	0.10	0.09	0.09	0.09	0.08	0.03	0.04	0.04	0.07
AKAZE [31]	**0.03**	**0.03**	**0.03**	0.04	0.03	0.05	0.03	0.03	0.02	0.02	0.02	0.03
FAST [33]	**0.03**	**0.03**	**0.03**	**0.01**	**0.02**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.02**
ORB [34]	0.86	0.89	0.88	0.93	0.84	0.91	0.91	0.83	0.14	0.14	0.14	0.68
F–H	21.19	21.29	24.18	31.66	16.56	36.42	16.18	25.67	10.32	9.71	10.36	20.32

Bold shows that nine algorithms correspond to the minimum running time of 10 different scenarios.

"Bridge", "Car", and "Instrument" were selected from the ImageNet dataset [45], and "Block image" [41], "Checkerboard image" [42], "House image" [41] are shown in Figure 8, to test the running time of the corner detection algorithm. The experiment is implemented in PyCharm-python3.8 with 2.60GHZ i7-9750H CPU and 8 GB memory. Each image runs 50 times, removes the longest and shortest running time, and then takes the average running time. Table 6 shows the comparison of the average running time. It can be seen from the table that the running time of this method is less than other algorithms; the running time is shortened by 17.37% compared with Harris algorithm especially.

6. Conclusion

In this paper, aiming at the low detection efficiency of Harris algorithm and the low detection accuracy of FAST algorithm, a corner detection method based on integrating FAST and Harris is proposed, and a corner detection model based on integrating FAST and Harris is established. First, in order to avoid the missing of image edge, the image is padded, and then the corner detection speed of FAST algorithm is improved by reducing the discrete detection points on the Bresenham circle. Then the candidate corner set is obtained by using the rough capture FAST algorithm with 7 * 7 as the matrix window; second, the accurate screening of candidate corners is carried out by accurately capturing Harris algorithm with 3 * 3 as matrix window. Finally, SAD is used for nonmaximum suppression to reduce corner redundancy.

Compared with Harris, Shi-Tomasi, Subpixel level, SIFT, SURF, AKAZE, FAST, and ORB, the proposed algorithm has the best performance in error detection rate and location error. Compared with Harris and FAST algorithms, the average error rate is reduced by 16.88% and 46.71%, respectively, and the average positioning error is 1.42 pixels. Compared with 8 algorithms, the performance of lossy JEPG compression has the best average repeatability, which is 0.56. Compared with Harris algorithm, the running time is reduced by 17.37%.

The data of the simulation show that an innovative corner detection model integrated the improved FAST and Harris. This method has better accuracy and robustness. The method can be used to image visual processing, motion estimation, image matching, visual tracking, and 3D scene reconstruction and so on, especially in the high quality and speed scene [46, 47].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Thanks are due to the Science and Technology Development Plan of Jilin Province for help in identifying collaborators for this work. This study was supported by the Science and Technology Development Plan of Jilin Province (20200401090GX).

References

[1] W. Wang, J. Li, H. Mo, and J. Chen, “Side information hybrid generation based on improved motion vector field,” *Multimedia Tools and Applications*, pp. 1–18, 2021.

[2] A. Bhat, “Makeup invariant face recognition using features from accelerated segment test and eigen vectors,” *International Journal of Image and Graphics*, vol. 17, no. 1, 1 page, 2017.

[3] Y. Chen, L. Liu, J. Tao et al., “The improved image inpainting algorithm via encoder and similarity constraint,” *The Visual Computer*, vol. 37, no. 7, pp. 1691–1705, 2021.

[4] Y. Chen, H. Zhang, L. Liu et al., “Research on image inpainting algorithm of improved total variation minimization method,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2021.

[5] F. Wu, J. Duan, S. Chen, Y. Ye, P. Ai, and Z. Yang, “Multi-target recognition of bananas and automatic positioning for the inflorescence axis cutting point,” *Frontiers of Plant Science*, vol. 12, Article ID 705021, 2021.

[6] X. Zhao, T. Zuo, and X. Hu, “OFM-SLAM: a visual semantic slam for dynamic indoor environments,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–16, Article ID 5538840, 2021.

[7] Y. Chen, L. Liu, J. Tao et al., “The image annotation algorithm using convolutional features from intermediate layer of deep learning,” *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4237–4261, 2021.

[8] B. Yu, R. Zhang, W. Chen, and J. Fang, “Graph neural network based model for multi-behavior session-based recommendation,” *GeoInformatica*, 2021.

[9] X. Cao, H. Yan, Z. Huang et al., “A multi-objective particle swarm optimization for trajectory planning of fruit picking manipulator,” *Agronomy*, vol. 11, no. 11, p. 2286, 2021.

[10] Y. Tang, M. Zhu, Z. Chen et al., “Seismic performance evaluation of recycled aggregate concrete-filled steel tubular columns with field strain detected via a novel mark-free vision method,” *Structures*, vol. 37, pp. 426–441, 2022.

[11] J. X. Pu, F. W. Song, and Q. Q. Leng, “Research on key technologies of 3D reconstruction based on SFM algorithm,” *Electronic Technology*, vol. 50, no. 06, pp. 36-37, 2021.

[12] Y. Chen, L. Liu, V. Phonevilay et al., “Image super-resolution reconstruction based on feature map attention mechanism,” *Applied Intelligence*, vol. 51, no. 7, pp. 4367–4380, 2021.

[13] Y. Tang, M. Chen, Y. Lin et al., “Vision-based three-dimensional reconstruction and monitoring of large-scale steel tubular structures,” *Advances in Civil Engineering*, vol. 2020, Article ID 1236021, 17 pages, 2020.

[14] H. Ou and J. Sun, “The multidimensional motion features of spatial depth feature maps: an effective motion information representation method for video-based action recognition,” *Mathematical Problems in Engineering*, vol. 2021, Article ID 6670087, 15 pages, 2021.

[15] D. Jia, Y. Li, S. Wu, and Y. Liu, “Linear correction and matching method for 3D line structure reconstruction,” *Mathematical Problems in Engineering*, vol. 2020, no. 5, pp. 1–11, Article ID 3597397, 2020.

[16] H. Li, K. Wang, K. Yang, R. Cheng, C. Wang, and L. Fei, “Unconstrained self-calibration of stereo camera on visually impaired assistance devices,” *Applied Optics*, vol. 58, no. 23, p. 6377, 2019.

[17] A. Lekova, D. Ryan, and R. Davidrajuh, “Fingers and gesture recognition with kinect v2 sensor,” *Inf. Technol. Control*, vol. 14, no. 3, pp. 24–30, 2017.

[18] C. Harris and M. Stephens, “A combined corner and edge detector,” *Alvey Vision Conference. [S.1.]*, University of Sheffield Printing Unit, Sheffield, UK, pp. 147–151, 1988.

[19] H. P. Moravec, “Towards automatic visual obstacle avoidance,” *Proc. Int. Joint Conf. on Artificial Intelligence*, p. 584, 1977.

[20] L. Dong, Y. Peng, and F. U. Li-Mei, “Circular Harris corner detection algorithm based on Sobel edge detection,” *Journal of Xi’an University of Science and Technology*, vol. 39, no. 2, pp. 374–380, 2019.

[21] S. Park, G. Kim, J. Park, and H. J. Yoo, “A1.5nJ/pixel super-resolution enhanced FAST corner detection processor for high accuracy AR,” *Proceedings of the ESSCIRC 2014 —40th European Solid State Circuits Conference*, vol. 11, no. 3, pp. 191–194, 2014.

[22] K. Mikolajczyk and K. Mikolajczyk, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[23] Z. Zhang and Z. Zhang, “Feature matching and target location algorithm based on Harris-improved LBP,” *Journal Of Jilin University (Science Edition)*, vol. 59, no. 3, pp. 568–576, 2021.

[24] Z. Yanling, C. Xixi, Q. Lu, and C. Pengfei, “A new fast corner detection method based on template matching,” *IOP*

Conference Series: Materials Science and Engineering, vol. 563, no. 5, Article ID 052093, 2019.

[25] Z. Wang, R. Li, Z. Shao et al., "Adaptive Harris corner detection algorithm based on iterative threshold," *Modern Physics Letters B*, vol. 31, no. 15, Article ID 1750181, 2017.

[26] J. Shi and C. Tomasi, "Good features to track," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 8, no. 6, 1994.

[27] C. Wang, H. U. Song, C. Feng, and C. Gao, "Deformation detection system of fuel assembly based on underwater binocular vision," *Journal of Applied Optics*, 2019.

[28] X. Ying and Z. Zhang, "Visual recognition of robot targets in complex state based on sub-pixel Harris corner," *E3S Web of Conferences*, vol. 233, Article ID 04029, 2021.

[29] D. G. Lowe, "Distinctive image features from scale-invariant k," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[30] T. L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[31] F. Pablo, "Alcantarilla, jesús nuevo and adrien bartoli, "fast Explicit diffusion for accelerated features in nonlinear scale spaces," *British Machine Vision Conference (BMVC)*, p. 9, 2013.

[32] L. Pallotta, G. Giunta, and C. Clemente, "Subpixel SAR image registration through parabolic interpolation of the 2-D cross correlation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4132–4144, 2020.

[33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *European Computer Vision*, Springer-Verlag, vol. 6, no. 5, pp. 430–443, Berlin, 2006.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, vol. 11, pp. 6–13, IEEE, Barcelona, Spain, 6 November 2011.

[35] N. Yao, T. C. Bai, and J. Chen, "Improved FAST corner detection based on Harris algorithm for Chinese characters," *Advanced Materials*, pp. 767–770, 2014.

[36] Z. Pan, X. Ma, L. Qian et al., "Three pulsars discovered by FAST in the globular cluster NGC 6517 with a pulsar candidate sifting code based on dispersion measure to signal-to-noise ratio plots," *Research in Astronomy and Astrophysics*, vol. 21, no. 06, pp. 186–192, 2021.

[37] X. Y. Chen, Y. Zhu, B. Xiao, Y. Dai, and J. Duan, "An algorithm for fast corner points extraction with reducing mistaken matching based on Harris operator," in *Proceedings of the 2017 5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017)*, pp. 562–567, Atlantis Press, 2017.

[38] J. Xu, X. Y. Chen, X. P. Song, and W. Gu, "An improved harris-FAST algorithm for underwater object corner detection," in *Proceedings Of the 27th Chinese Conference On Control And Decision (Volume II)*, pp. 2433–2437, 2015.

[39] J. Wang, X. U. Xiaofeng, M. Dong, P. Sun, and M. Chen, "Relative pose estimation method of monocular mobile robot," *Journal of Applied Optics*, 2019.

[40] M. Wang and T. Hong, "An Improved Image Edge Detection Method Based on Scharr Operator," in *Proceedings of the The 2020 3rd International Symposium on Advanced Intelligent Informatics (SAIN)*, Yogyakarta, Indonesia, 3 November 2020.

[41] F. Mokhtarian, "Image Corner Detection through Curvature Scale Space," 2001, http://www.ee.surrey.ac.Uk/CVSSP/demos/corners/.

[42] F. Tong, "Research on corner detection accuracy evaluation method," *Software Guide*, vol. 20, no. 5, pp. 173–177, 2021.

[43] M. Awrangjeb, G. J. Lu, and M. Murshed, "An affine resilient curvature scale-space corner detector," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. I-1233–I-1236, IEEE, Honolulu, HI, USA, 15 April 2007.

[44] https://cocodataset.org/#download.

[45] https://image-net.org/.

[46] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[47] Y. Wang, H. Lv, Y. Li, and Z. Yang, "Optimization of corner detection algorithm for video stream based fast," in *Proceedings of the 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pp. 479–483, IEEE, Changchun, China, 23September 2021.