



CCDA: a concise corner detection algorithm

Zhiyong Peng^{1,2,3} · Jun Wu¹ · Guoliang Fan⁴

Received: 3 December 2017 / Revised: 15 April 2019 / Accepted: 23 May 2019 / Published online: 11 June 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In this article, the authors propose a concise corner detection algorithm, which is called CCDA. A cascade classifier concept is used to derive a corner detector, which can quickly discard the most non-corner pixels. The ruler of gradient direction is used to get the corner, which can avoid the influence of the light change. The method of second derivative non-maximum suppression is used to get the location of the corner and can get the exact corner point. As a result, CCDA is compare-tested with classical corner detection algorithms by using the same images which include synthetic corner patterns and real images. The result shows that CCDA has a similar speed to the FAST algorithm and better accuracy and robustness than the HARRIS algorithm.

Keywords Corner detection · Cascade classifier · Gradient direction · Non-maximum suppression

1 Introduction

In many computer vision applications, keypoint detection is indispensable, and it can be used in applications such as object detection [1], 3D reconstruction [2], scene understanding [3] and so on. A corner can be used as an keypoint to understand the context or extract features. In this paper, we present a fast, accurate and robust approach to detect corners based on the cascade classifier concept [4]. Finally, it is proved that the CCDA has similar speed to the FAST algorithm and better accuracy and robustness than the HARRIS algorithm as evidenced by many experiments.

The corner is generally defined as the local maximum of the curve's absolute curvature value [5]. Corner detection is

very complex, because there is not a closed solution to corner detection. There are many researchers who have proposed some classical corner detection algorithms. In [6], a method of corner estimation using the affine morphological scale space was proposed. The HARRIS corner detection (HCD) [7] algorithm is one of the most widely used for high accuracy and stability. While the FAST corner detection algorithm [8] is known for efficiency compute.

HARRIS was proposed by Moravec [9] and was further improved by Harris and Stephens [10]. Harris and Stephens filter out the noise by using a circular Gaussian window instead of a rectangular window and eliminate HARRIS's anisotropic response. Furthermore, Harris matrix A is used to find the direction of the fastest and the lowest change, which is regarded as feature orientation. The directional derivative values are used to identify the features of corners.

Afterward, the HARRIS algorithm was improved by Shi and Tomasi. The selection criteria was changed in the Shi-Tomasi corner detector (GFTT) [11]. The minimum eigenvalue was used to check if the pixel is a corner or not. In the GFTT detector, R is defined as: $R = \min(\lambda_1, \lambda_2)$, where λ_1, λ_2 are eigenvalues of Harris matrix A . If R is greater than the threshold value, this region is “accepted” as a corner.

The FAST algorithm is related with the local binary pattern LBP [12] and is derived from SUSAN [13,14]. FAST uses binary comparison, which compares each pixel in a circular pattern against the center pixel. The result descriptor is

✉ Jun Wu
wujun93161@163.com
Zhiyong Peng
pzy@guet.edu.cn

¹ College of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China
² Guangxi Experiment Center of Information Science, Guilin 541004, China
³ Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin 541004, China
⁴ School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma 74074, USA

stored as a contiguous bit vector. The corner is requested to be a connected set. Because FAST is efficient and accurate, it is used in the BRISK and ORB binary feature detection algorithm.

In the BRISK algorithm [15], the FAST detector is improved, as it can adapt to scale space. Corners are used as keypoints and are detected in octave layers of the image pyramid and in layers in between. The location and the scale of each keypoint is obtained in the continuous domain by quadratic function fitting. ORB [16] is based on BRIEF [17,18] and adds rotational invariance to BRIEF by determining corner orientation using FAST9, then is followed by a Harris corner metric to sort the keypoints. The corner orientation is refined by intensity centroids which use Rosin's method [19]. The FAST, Harris, and Rosin processing are done at each level of an image pyramid scale.

These classical corner detection algorithms have respective advantages and disadvantages. Based on OpenCV, these algorithms are tested by using the synthetic corner images in Sect. 4. From the results of the test, it can be found that the HARRIS algorithm found the most corner points and has the highest accuracy rate, but it needs a long computation time. The performance of GFTT is similar to HARRIS. The FAST algorithm is the fastest, but the accuracy rate is low. In the BRISK algorithm, the accuracy of FAST has been improved, but the amount of corner points is decreased, many corners cannot be found. This paper proposes a concise novel method for corner point detection.

The main contribution of this paper is that a novel corner detection method is proposed by using the cascade classifier concept to quickly discard most non-corner pixels. This paper uses gradient direction rather than the gradient value to distinguish edge pixels and corner pixels, thus reducing the influence of the change in light. This paper uses second derivative non-maximum suppression to get exact corner point location. CCDA not only has fast speed, but also has better accuracy and robustness than HARRIS algorithm. This paper also experimentally studies the performance of algorithms by synthetic and real corner images and compares CCDA with classical corner detection algorithms. The result proves the new method has a similar speed to the FAST algorithm and better accuracy and robustness than classical algorithms.

The remaining of this paper is organized as follows. Section 2 discusses some related works. Section 3 provides fine details of CCDA. Section 4 presents some experiments by using synthetic and real images, in particular, it compares CCDA with classical corner detection algorithms. Section 5 analyzes the influence of different parameters for the result. Section 6 summarizes the major findings and concludes of the paper.

2 Related work

In the HARRIS algorithm, the fastest and the lowest change directions are found by using Harris matrix A , which is shown as formula (1).

$$A = \sum_{u,v} w(u,v) \begin{bmatrix} I_x^2(u,v) & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y^2(u,v) \end{bmatrix}. \quad (1)$$

I_x and I_y are the partial derivatives of the x direction and the y direction. The eigenvalues of matrix A can be regarded as a rotational invariant description for the windowed image region. But the eigenvalue decomposition of matrix A has a complex calculation.

Commonly, a corner has a larger curvature on the curve than other points. Corners are often preferred over edge pixels so detectors can use maxima and minima points to get the corner in the local edge pixels. Edge pixels usually can be found by gradient magnitude. Gradient magnitude M is the quadratic sum of first derivative of the pixels in the local region.

$$M = (\partial f(x,y)/\partial x)^2 + (\partial f(x,y)/\partial y)^2. \quad (2)$$

At the same time, the corner is a structure with angular orientation, which connects with gradient orientation in the local region. It can be judged if the corner is true or not by gradient orientation. Gradient direction θ is the direction of the angle, and it is in the range of $+\pi$ to $-\pi$.

$$\theta = \tan^{-1}((\partial f(x,y)/\partial y)/(\partial f(x,y)/\partial x)). \quad (3)$$

The gradient direction of the corner is the angular orientation. The adjacent pixels of the corner point, which are on the vertical direction of the gradient direction, have different gradient direction than the corner point. The edge pixel has the same gradient direction as the corner point. So we can distinguish a corner from edge pixels in accordance to the change of gradient direction.

Usually, derivative operator includes Roberts operator [20], Prewitt operator [21], and Sobel operator [22]. The Sobel operator uses different weight value according to the distance and can effectively suppress noise, which is shown in Fig. 1. The Sobel operator can be used to effectively get possible edge pixels in the image.

The Laplacian operator, as second derivative method, can be used to find the derivative or maximum change rate in a pixel area. The corner is usually the maximum change value in the region. The Laplacian operator is easy to be influenced by noise. Laplacian of Gaussian (LOG) combines Laplacian and Gaussian, which has been processed using a Gaussian

-1	0	1
-2	0	2
-1	0	1

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

Fig. 1 Sobel operator: **a** it can be use to detect vertical gradient, **b** it can be use to detect horizontal gradient

Fig. 2 LOG operator

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

			p_{15}	p_1	p_2			
	p_{15}		1	1	1		p_3	
		1				1		
p_{14}	1						1	p_4
p_{13}	1			p_0			0	p_5
p_{12}	1						0	p_6
		1					0	
	p_{11}		1	0	0		p_7	
			p_{10}	p_9	p_8			

Fig. 3 The FAST detector with a 16-element circular sampling pattern grid. Each pixel in the circular is compared with the center pixel to yield a binary value 1 or 0

smoothing kernel to focus edge energy [23]. The approximate LOG operator is shown in Fig. 2.

In the FAST algorithm, the corner is requested to be a connected set of pixels in a circular. The connected region size is commonly 9 or 10 out of a possible 16, referred to as FAST9 and FAST10. The FAST algorithm compares each pixel in a circular pattern with the center pixel. If a pixel is greater than the center pixel, the result is set as “1”; otherwise, it is set as “0”. If it is a corner, the pixels in the circular figure are two connected sets of “1” and “0”, as shown in Fig. 3. If the algorithm just uses FAST9 or FAST10 to get corner, it is not robust. But if the pixel doesn’t meet the condition which has two connected sets, it is definitely not a corner. So we can use this ruler to remove non-corner pixels.

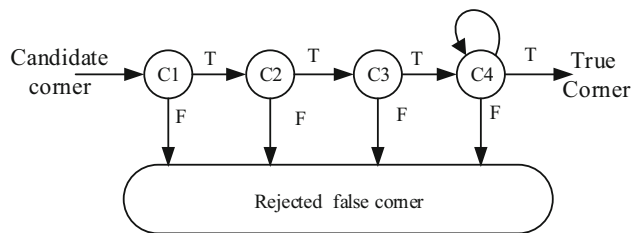


Fig. 4 The structure of algorithm of CCDA

3 CCDA corner estimation algorithm

3.1 The structure of algorithm

Authors adopt the cascade classifier concept to detect corner points, which can reduce the computation time by discarding the most non-corners quickly. The basic idea of the cascade classifier is to reject the negative sample as early as possible through the cascade method, because most pixels are negative samples (non-corners) in a image. The final strong corner decision process is divided into multi-level judgments, the judging conditions of every level are set. As a result, the negative samples can be removed early. The structure of the algorithm is shown in Fig. 4.

In the first level, to set the condition C_1 , the algorithm uses a gradient operator to get possible edge pixels. Many background pixels are discarded quickly by condition C_1 . Commonly, the pixels which are discarded go beyond 85% in the first-level detector.

In the second level, to set the condition C_2 , the algorithm uses a Laplace operator to get the local extrema in the possible edge pixels. Many non-corner edge pixels are discarded quickly by condition C_2 . Commonly, the pixels, which are discarded, go beyond 90% until the second-level detector is used. At the same time, the location of possible corner pixels can be found.

In the third level, to set the condition C_3 , the algorithm uses the change ruler of gradient direction to get candidate corners. The change ruler of gradient direction states that the adjacent pixels on the vertical direction of gradient must have different gradient direction than the corner.

In the fourth level, to set the condition C_4 , the algorithm uses the distribution ruler of gray value to get the true corners. The true corners need to meet the ruler where pixels in the circular must be two connected sets of “1” and “0”. This paper uses this method to detect the corner shape and get the true corner pixels.

3.2 Each level detection algorithm

The purpose of the first-level detector is to get possible edge pixels. The purpose of the second-level detector is to get the location of possible corner pixels. The purpose of the third-

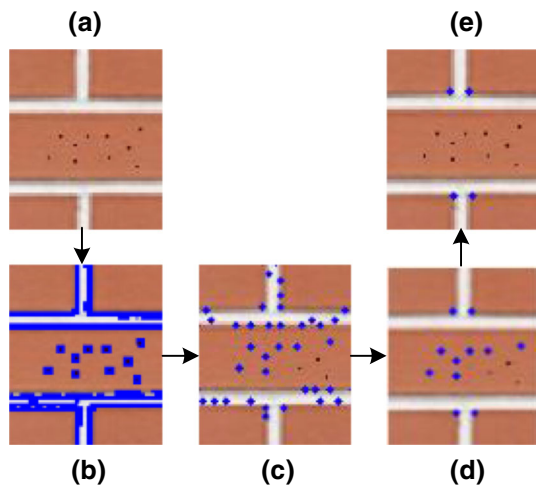


Fig. 5 The results of every level in CCDA: **a** the original image, **b** the possible edge pixels are found by the first level, **c** the accurate location of candidate corner are found by the second level, **d** the refined candidate corners are found by the third level, **e** the true corners are found by the fourth level

level detector is to remove edge pixels which is not corner. The purpose of the fourth-level detector is to remove random non-corner noise pixels. The processing result of each level is shown in Fig. 5.

3.2.1 The first-level detector

In this paper, the Sobel gradient operator is used to get the possible edge pixels by calculating the gradient value in the x and y directions.

$$\nabla_x = \frac{\partial f(x, y)}{\partial x}, \nabla_y = \frac{\partial f(x, y)}{\partial y}. \tag{4}$$

In the calculation of Gradient Magnitude, the absolute value calculation is used to replace square calculation for reducing amount of calculation.

$$\nabla(x, y) = |\nabla_x| + |\nabla_y|. \tag{5}$$

If pixel (x, y) meets the condition $C_1 : \nabla(x, y) > D_1$, the pixel is possible edge pixels; otherwise, it is discarded. D_1 is a threshold value.

3.2.2 The second-level detector

The approximate LOG operator is used to get $\nabla^2(x, y)$ and then calculate the local extrema in the possible edge pixels.

$$\nabla^2(x_0, y_0) = \max(|\nabla^2(x, y)|)_w. \tag{6}$$

w is the window, $\max()$ is to get the max value of possible edge pixels in the w .

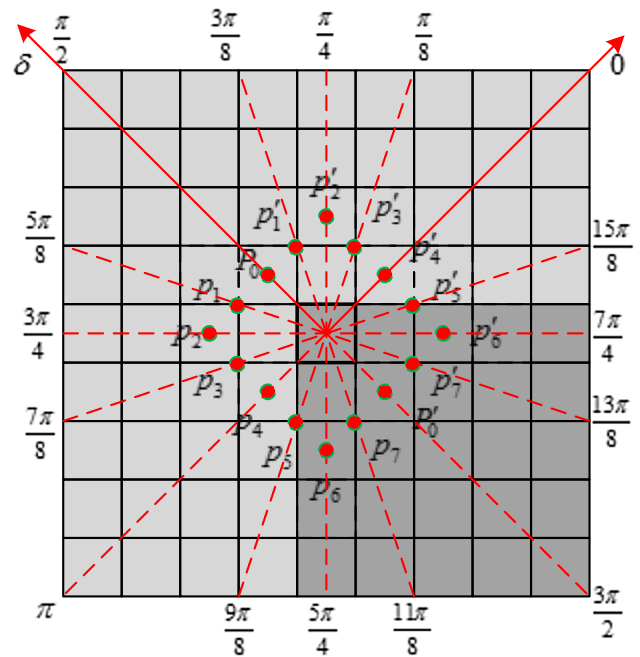


Fig. 6 The polar coordinate of candidate corner and the distribution of detecting pixels. $P_0, P'_0, P_1, P'_1, P_2, P'_2, P_3, P'_3, P_4, P'_4, P_5, P'_5, P_6, P'_6, P_7, P'_7$ are used to judge the shape of corner

The accurate location of candidate corner is found by the non-maxima suppression method. The process of calculation is:

First, the formula (6) is used to get the max value $\nabla^2(x_0, y_0)$ of the possible edge pixels in the window w .

Second, it is judged whether (x_0, y_0) is the center of the window or not. If the result is true, jump to the fourth step; otherwise, continue to do the third step.

Third, the center of the window w is removed to (x_0, y_0) , and jump to the first step.

Fourth, the candidate corner (x_0, y_0) is got.

3.2.3 The third-level detector

The gradient direction of a candidate corner (x_0, y_0) is calculated by ∇_x, ∇_y .

$$\delta = \arctan(\nabla_y/\nabla_x). \tag{7}$$

For each candidate corner, there is a polar coordinate (θ, ρ) by setting the gradient direction calculated as $\delta = \pi/2$ and candidate corner as $\rho = 0$, which is shown in Fig. 6.

In the direction of $\theta = 0$ and π , getting two points P_4 and P'_4 with the condition:

$$|x_0 - x'| + |y_0 - y'| = 2. \tag{8}$$

The gradient value ∇_x and ∇_y of P_4 and P'_4 are calculated by bilinear interpolation, and then, their gradient directions

δ_0 and δ_π are calculated. If δ_0 and δ_π meet the condition $C_3 : |\delta_0 - \delta| > D_3 \cap |\delta_\pi - \delta| > D_3$, the center point (x_0, y_0) is possibly a corner pixel; otherwise, it is discarded.

3.2.4 The fourth-level detector

In the end, the true corners are detected in the candidate corners using the Polar coordinate. We choose 16 points $\rho(x', y')$ around the candidate corner: $\theta = 0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}, \pi, \frac{9\pi}{8}, \frac{5\pi}{4}, \frac{11\pi}{8}, \frac{3\pi}{2}, \frac{13\pi}{8}, \frac{7\pi}{4}, \frac{15\pi}{8}$. The distance ρ is required to meet the condition: $|x_0 - x'| + |y_0 - y'| = 2$. As shown in Fig. 6.

The gray value of 16 points is used to compare with the threshold value $F(I(P_0), I(P'_0))$, respectively. The true corner is required to meet the condition C_4 : the pixels in a circular are divided into two connected sets of “1” and “0”, around P_0 which has a connected set of “1”, around P'_0 has a connected set of “0”.

First, two points P_0 and P'_0 are on the line $\theta = \pi/2$ and $3\pi/2$. The gray value $I(P_0)$ and $I(P'_0)$ are obtained via bilinear interpolation. The corner is required to meet the condition:

$$\begin{cases} I(P_0) > D_4 \\ I(P'_0) < D_4 \end{cases}, D_4 = \frac{I(P_0) + I(P'_0)}{2}. \quad (9)$$

Second, two by two detecting the rest of points orderly from $P_0 : \pi/2$ to $P'_0 : 3\pi/2$ in the left and right direction. The order is $[P_1, P'_1] \rightarrow [P_2, P'_2] \rightarrow [P_3, P'_3] \rightarrow [P_4, P'_4] \rightarrow [P_5, P'_5] \rightarrow [P_6, P'_6] \rightarrow [P_7, P'_7]$. If the algorithm finds point P_i or P'_i meet $I(P_i) < D_4$ or $I(P'_i) < D_4$, the rest of points ($[P_j, P'_j]$, all $j > i$) are required to meet the condition: $I(P_j) < D_4$ and $I(P'_j) < D_4$

4 Experimental test

In this paper, some compare-tested experiments are designed to illustrate the speed, accuracy and robustness of CCDA. The accuracy and speed are tested by using Scott Krig’s synthetic corner images [24]. The robustness is tested by using noise synthetic corner images. In the end, CCDA is tested by using real images.

4.1 Accuracy and speed test

The set of synthetic corner images is comprised by image units. The original image unit and test results are shown in Fig. 7. The synthetic corner image unit contains 54 unique patterns. The total of 8×12 image units of the 54 patterns fit within the 1024×1024 image. The total amount of corner patterns is $8 \times 12 = 5184$. Each pattern is arranged on a grid

of 14×14 pixel rectangles. Gray values are $0x40$ and $0xC0$. In order to eliminate the influence of computer random state for test time, we adopt the method of continuous testing 10 times then we get the average. If the grid distance between the detected corner and true corner is more than 1 pixel, the detected corner is considered imprecise. Data results of the classical corner detection algorithms and CCDA are illustrated in the Table 1.

Configuration parameters are:

BRISK: octaves = 3, threshold = 30.

FAST: FAST10, threshold = 10, nonMaximalSuppression = TRUE.

HARRIS: maxCorners = 60,000 (to capture all corners), qualityLevel = 1.0, minDistance = 1, blockSize = 3, useHarrisDetector = TRUE, k = 0.04.

GFTT: maxCorners = 60,000 (to capture all corners), qualityLevel = 0.01, minDistance = 1.0, blockSize = 3, useHarrisDetector = FALSE, k = 0.04.

CCDA: k1 = 1.2, D3 = 20.

The results show CCDA’s speed is similar with FAST algorithm, in 12,347 milliseconds, which finished corner detection of image with 1024×1024 resolution and found 36,384 corners. The accuracy rate of CCDA is the best in the tested algorithms, achieving accuracy rate is 97.19%.

4.2 Robustness test

First, synthetic corner images are added with salt-and-pepper noises, and the noise density is $D = 0.05$. We use the same value of parameters as Sect. 4.1 to test salt-and-pepper noise image. The salt-and-pepper noise image and test results are shown in the Fig. 8. As before, we adopt the method of continuous testing 10 times, and then, we get the average with the same configuration parameters. Data results of the classical corner detection algorithms and CCDA are illustrated in Table 2. Results show the performance of all algorithms have dropped for noise. But the speed of CCDA still is very fast, in 8954 milliseconds, which finished corner detection of image with 1024×1024 resolution and found 29,012 corners. The accuracy rate of CCDA is better than other algorithms. The accuracy rate of CCDA is 95.04%.

Then, synthetic corner images are added with Gaussian noise, where the mean is $M = 0$ and the variance is $V = 0.01$. We use the same value of parameters as Sect. 4.1 to test Gaussian noise image. The Gaussian noise image and testing results are shown in Fig. 9. As before, we adopt the method of continuous testing 10 times and then get the average with the same configuration parameters. Data results of the classical corner detection algorithms and CCDA are illustrated in Table 3. Results show that the performance of all

Fig. 7 The test results of original synthetic corners image

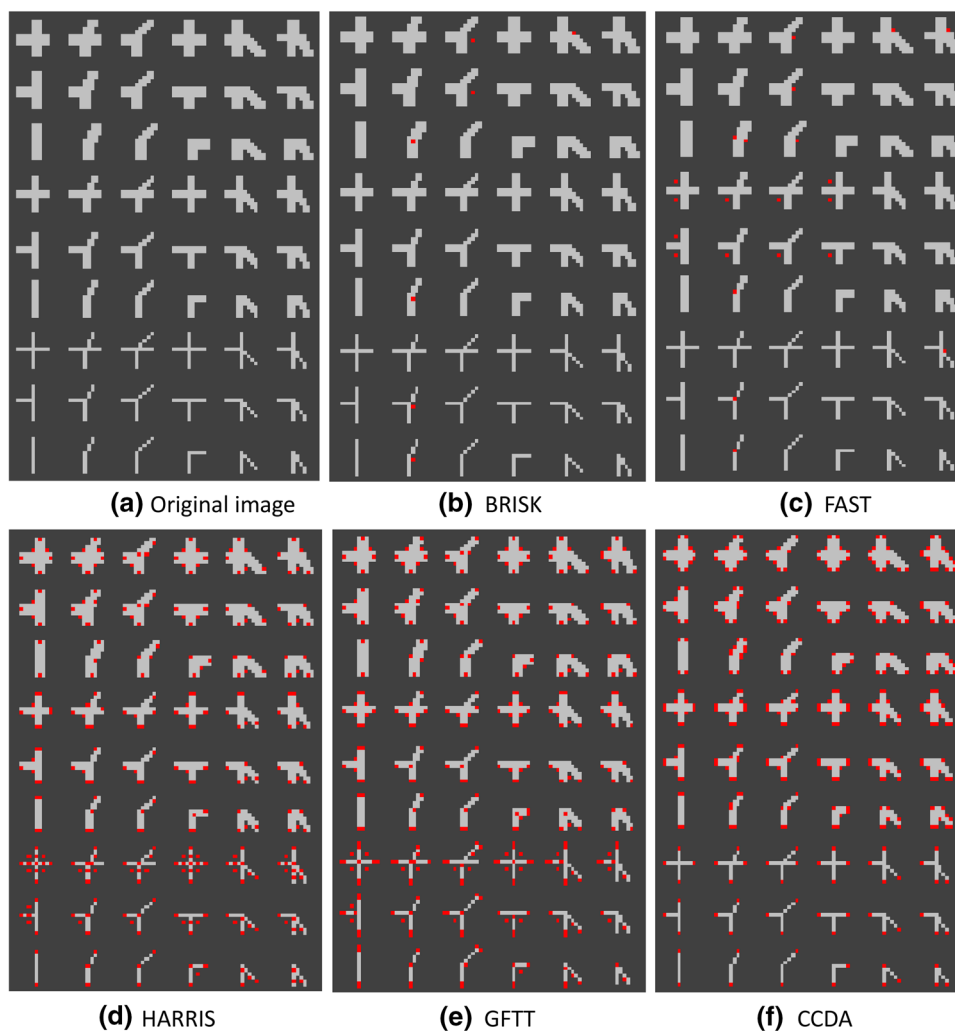


Table 1 The data report of testing for original synthetic corners image

Detector	Corner Points	ime (ms)	Time/point(ms)	Incorrect points	Precision rate (%)
BRISK	672	3403	5.06	192	71.43
FAST	2016	737	0.37	1056	47.62
HARRIS	29,760	303,624	10.2	4032	86.45
GFTT	32,597	309,335	9.49	6144	81.15
CCDA	36,384	12,347	0.34	1024	97.19

algorithms are not good for noise. The corner number of CCDA has reduced to 11,269, but the speed is still very fast, in 5017 milliseconds the algorithm finished corner detection of image with 1024×1024 resolution. The accuracy rate of CCDA still is best in the tested algorithms. The accuracy rate of CCDA is 88.9%.

4.3 Repeatability test

To evaluate the interest point detection ability of CCDA, we measure repeatability on the Edward Rosten [25] dataset. The image dataset includes Maze and Bas-relief which are

shown in the Fig. 10. The resolution of image is 768×576 . Maze dataset (Fig. 10a) has an abundance of textural features and geometric features, furthermore, it has a heavy projection warp. Bas-relief dataset (Fig. 10b) has significant relief feature, furthermore, the feature has variation from different viewpoints. We use a repeating distance of 3 pixels and compare new algorithm to FAST, HARRIS, DOG and GFTT, all implemented using OpenCV. Results are shown in the Fig. 11. It can be found that new algorithm's repeatability is best in the test algorithms for the different number of feature point. The new algorithm has obvious advantages in Maze

Fig. 8 The test results of salt-and-pepper noise images

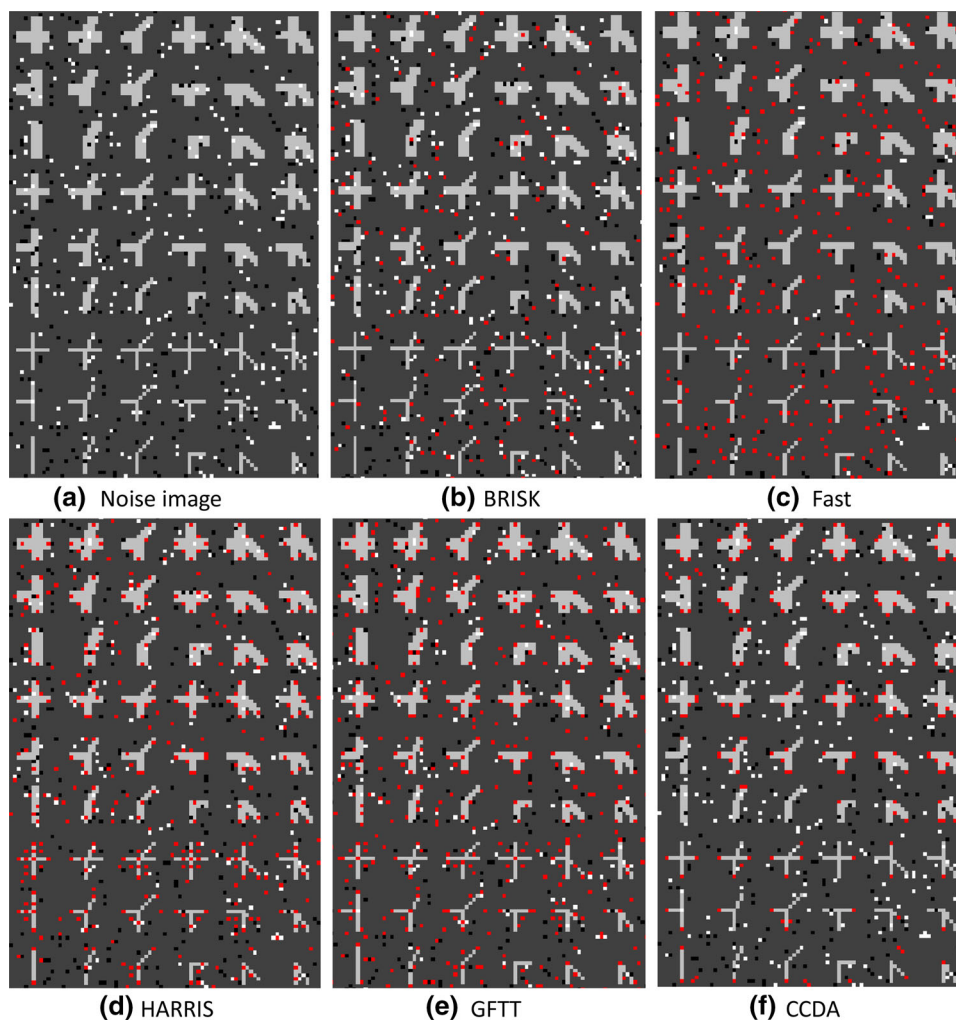


Table 2 The data report of salt-and-pepper noise images

Detector	Corner points	ime (ms)	Time/point(ms)	Incorrect points	Precision rate (%)
BRISK	13,178	4501	0.34	12, 122	8
FAST	34,809	2271	0.07	31, 449	9.7
HARRIS	38,979	312, 010	8.0	11, 040	71.7
GFTT	43,775	317, 689	7.26	24, 095	44.96
CCDA	29,012	8954	0.31	1440	95.04

dataset because the images have clear and abundant corner points.

To evaluate the influence of illumination changes for detection result of CCDA, we measure the number of interest point and repeatability on K. Mikolajczyk’s leuven dataset [26] and HPatches dataset [27]. The test dataset includes 14 scenes of HPatches dataset and K. Mikolajczyk’s leuven dataset, which have obvious 6 illumination grades from bright to dusky. We use fixed values of parameters to test the different illumination images and compared new algorithm to FAST, HARRIS, SIFT and GFTT, all implemented using OpenCV. To get similar number of interest point for differ-

ent algorithm in the brightest image, the value of parameters are set as: in FAST, threshold = 20; in HARRIS, $k = 0.03$; in GFTT, $k = 0.03$; the other parameters are same with Sect. 4.1. SIFT use the default parameters. The average number of interest point in each illumination grade is shown in Fig. 12a. We use a repeating distance of 3 pixels to test the interest point repeatability of the brightest image in the other images. The result is shown in Fig. 12b. It can be found that the number of interest point hasn’t reduced when the average gray of image reduce by using fixed values of parameters in CCDA. The number of interest point has fast reduced when average gray of image reduce with fixed parameters in other

Fig. 9 The testing results of Gaussian noise images

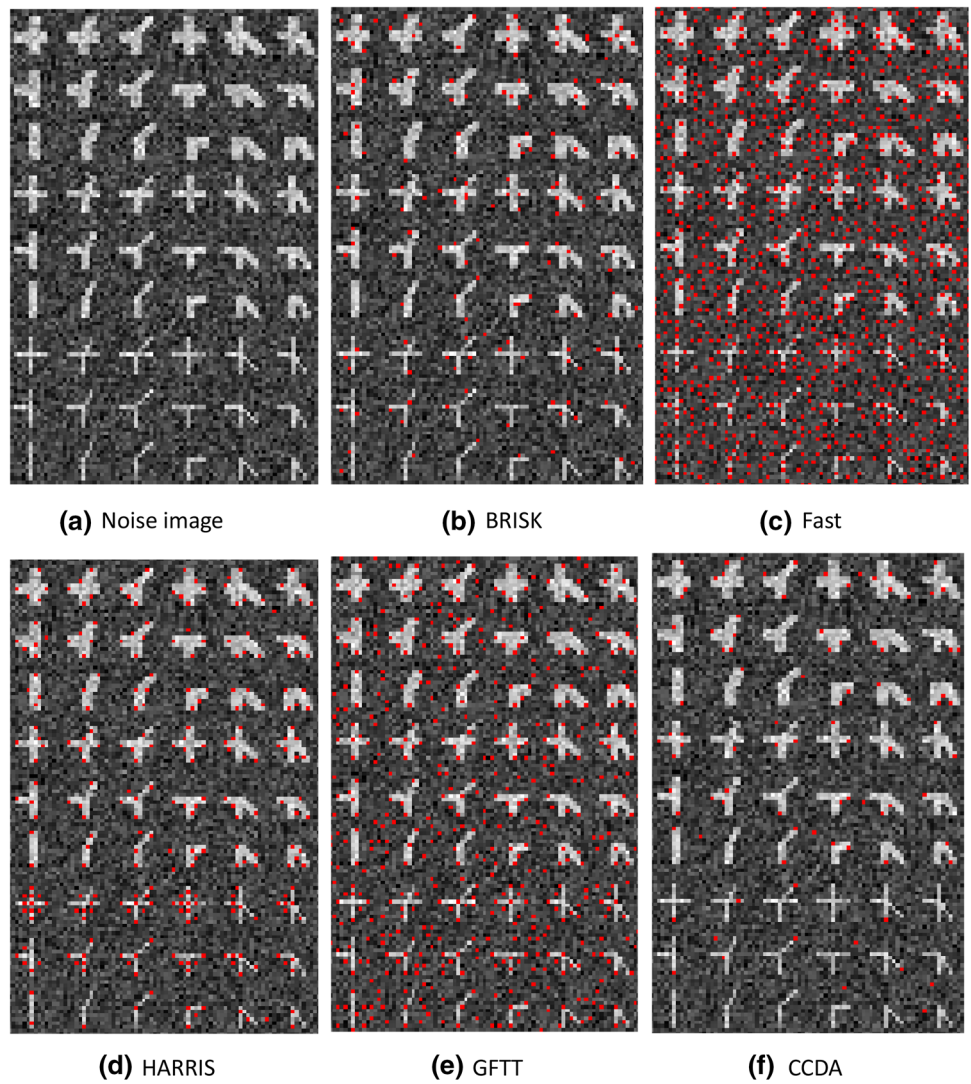


Table 3 The data report of Gaussian noise images

Detector	Corner points	ime (ms)	Time/point(ms)	Incorrect points	Precision rate (%)
BRISK	14,742	4529	0.31	3936	73.3
FAST	94,016	5224	0.06	80,384	14.5
HARRIS	27,425	305,178	11.13	6336	76.9
GFTT	51,173	311,732	6.09	30,053	41.27
CCDA	11,269	5017	0.45	1251	88.9

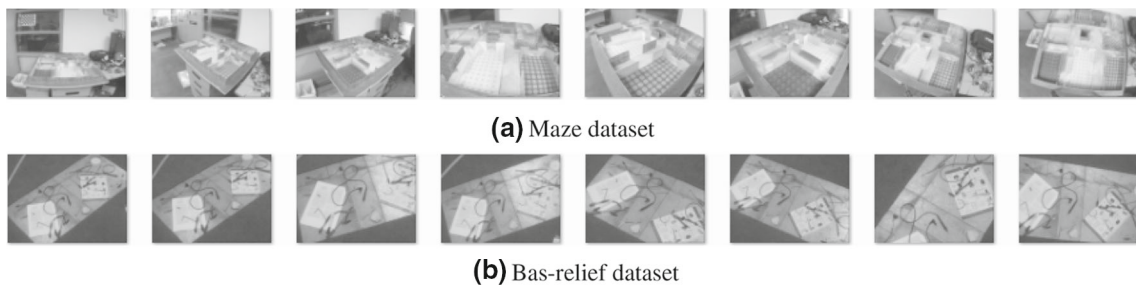


Fig. 10 Edward Rosten’s repeatability test datasets

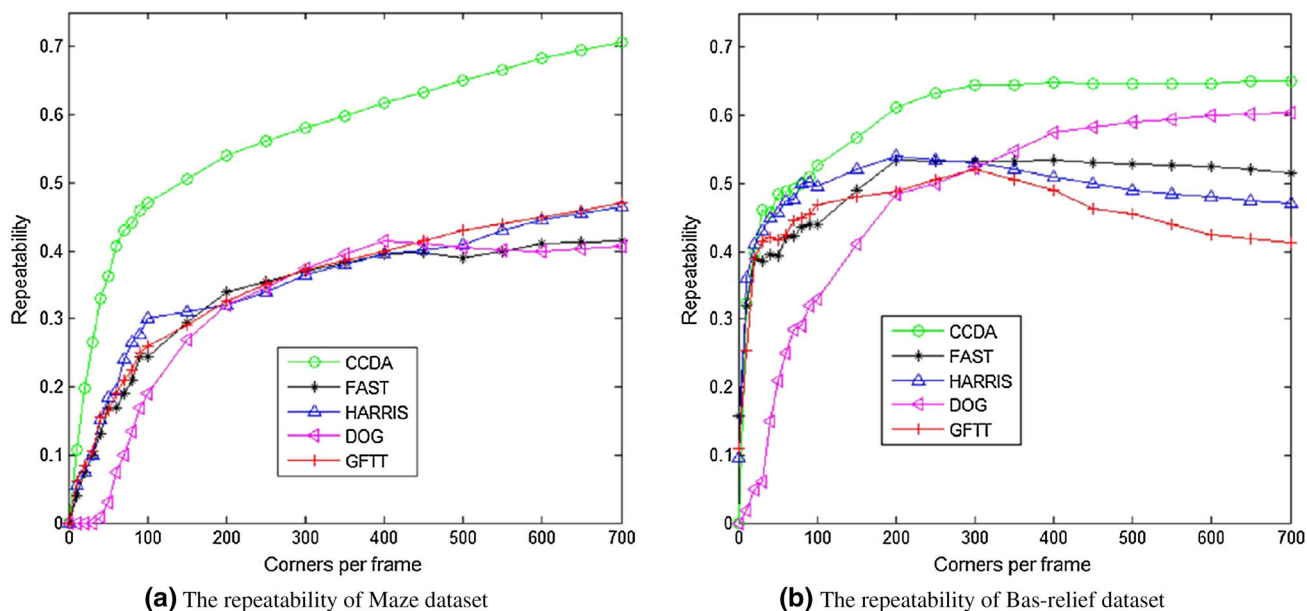


Fig. 11 The test results of repeatability on Edward Rosten's datasets

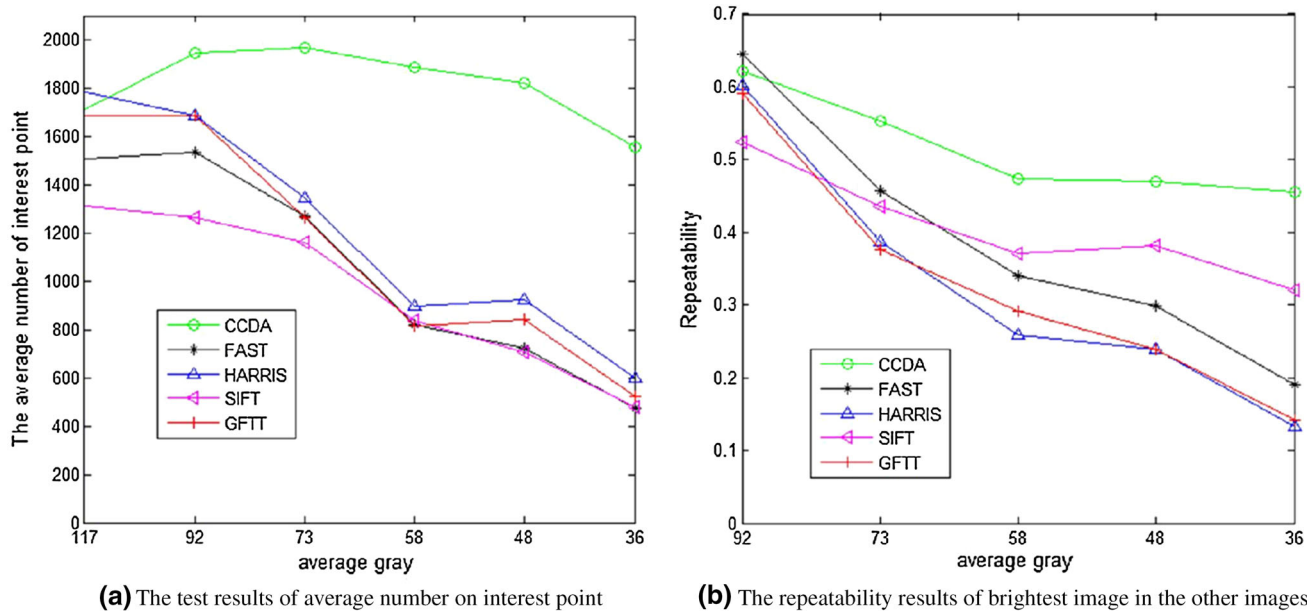


Fig. 12 The test results of different illumination grade images

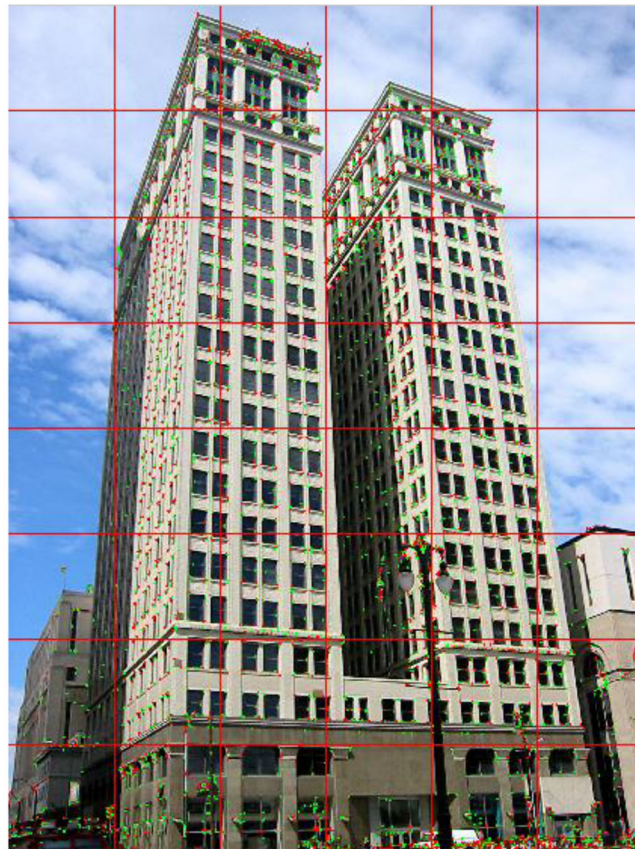
algorithms. At the same time, the new algorithm's repeatability of brightest in the other images hasn't quickly reduced as other algorithms. So the result shows that the influence of the illumination change for CCDA is weaker than others, and the new algorithm can have robust parameters for different illumination image.

4.4 Real image test

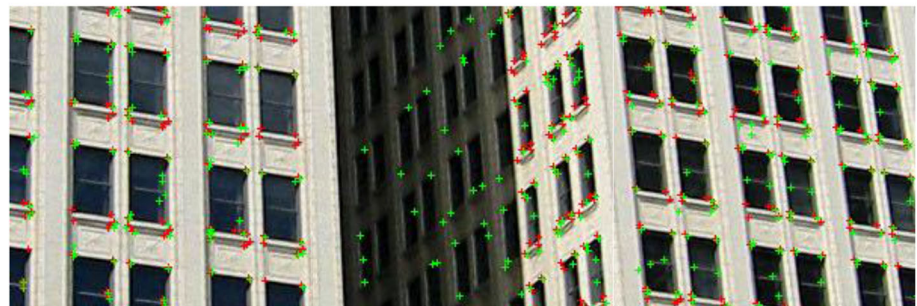
It can be found that the performance of the HARRIS algorithm is the best in classical algorithms by previous tests. So

we compare-test the HARRIS algorithm and CCDA algorithm and show results in the same real image, which is shown in Fig. 13. The green points are the corner points of CCDA, and the red points are the corner points of HARRIS. For a more clearly shown the result, we divide the image into image grids and zoom in on the image grids. From the result, it can be found that the detection result of CCDA is similar with HARRIS in the bright region. In the dark shadow region, HARRIS is almost unavailable, but CCDA can available detect the corner. It shows that CCDA can available avoid the influence of illumination. Furthermore, CCDA fin-

Fig. 13 The test result of real image. The green points are the corner points of CCDA, the red points are the corner points of HARRIS. **a** the result, **b** the middle grids of image are zoomed in



(a)



(b)

ished the test in 9617 milliseconds and got 4193 corners in image with 1200×1600 resolution, comparatively HARRIS needs 556,958 milliseconds to finish the test and got 4809 corners.

5 Results and discussion

It can be found that HARRIS is the most accurate and robust in the classical corner detection algorithms, but HARRIS is very complex and need more computation time; furthermore, HARRIS is almost unavailable in the dark region with low gray value. FAST method has a fast detection speed, but many corners can't be found and it isn't accurate

enough; furthermore, it is very easy to be influenced by noise. The speed of CCDA is the similar to FAST, but the robustness and accuracy is better than HARRIS.

The CCDA algorithm has 4 main judge conditions: C_1 , C_2 , C_3 , and C_4 .

Condition C_1 is used to get possible edge pixels by gradient. It is a very weak condition and commonly all edge pixels can meet the condition. In the paper, the threshold D_1 is connected with the average gray value.

$$D_1 = k_1 \overline{I(x, y)}. \quad (10)$$

k_1 is a constant which can be selected empirically, In practice its value lies between 1.1 and 1.5. $\overline{I(x, y)}$ is the average gray value of image.

Condition C_2 is used to get possible corners. We didn't use threshold value to detect possible corners, but by a moving window to get the local maximum, and to get possible corners. This method can avoid the influence of the change of light, and this method is available for an image with different brightness. CCDA needs to set a suitable window. If the window's size is too large, some corners will be lost; on the contrary, the calculated amount will increase. In the paper, the window's size is set to 5×5 ;

Condition C_3 is used to remove edge pixels which aren't corners. We remove the edge pixels which are not a corner according to the change in the gradient direction. This method can also avoid the influence of the change of light. The change in gradient direction is smaller when the angle of the corner is larger. In practice, angle range of a corner is requested from 0° to 160° . So the value of threshold D_3 is set to 20.

Condition C_4 is used to remove noise pixels. If a point is a noise point, the gray values will randomly distribute in the adjacent region. The threshold D_4 is requested to be different in the different gray value region, and it is connected with local gray value. So we use the average value of adjacent pixels which lie in the gradient direction and the opposite gradient direction.

In the test, it can be found that CCDA can accurately and quickly detect corners in the image. At the same time, CCDA can be applied to an image with a different gray level.

6 Conclusion

In this paper, a novel concise corner detection method CCDA is proposed. The main advantage of CCDA is that it is fast, accurate and robust; moreover, CCDA is not sensitive to the change of light. CCDA uses the cascade method to reject non-corner pixels as early as possible. Many non-corner pixels are discarded quickly; thus, it is fast. In this paper, first, the gradient operator is used to quickly get possible edge pixels in the image. Second, the LOG operator non-maxima suppression is used to get the accurate candidate-corner location in the edge pixels. Third, using the change in gradient direction to remove edge pixels which are not a corner, it can effectively reduce the influence of the gray value on the result. Fourth, the pixels are in adjacent circular, which are required to be two connected sets of "1" and "0". We can remove the influence of random non-corner noise pixels by using this method. Finally, getting the true corner pixels. We present a detailed experimental study of the speed, accuracy and robustness by the set of synthetic corners with different shapes and real images. The test results show that CCDA is not only fast but also has high accuracy and robustness. In further work, CCDA can be improved and used to detect interest points and can be widely applied to real-time image features detection.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (41761087) and Guangxi Natural Science Foundation (2017GXNSFAA198162), by Foundation of Guangxi Experiment Center of Information Science (YB1414), by Innovation Project of Guangxi Graduate Education (YCBZ2017051), by Guangxi College's emphasis laboratory foster base for optoelectronics information (handling) Project (GD18108), and by the study abroad program for graduate student of Guilin University of Electronic Technology.

References

1. Yan, C., Xie, H., Chen, J., Zha, Z., Hao, X., Zhang, Y., Dai, Q.: A fast Uyghur text detector for complex background images. *IEEE Trans. Multimed.* **20**(12), 3389–3398 (2018)
2. Zhang, S., Liu, W.: Single image 3D reconstruction based on control point grid. *Multimed. Tools Appl.* **77**(3), 1–19 (2018)
3. Yan, C., Li, L., Zhang, C., Liu, B., Zhang, Yongdong, Dai, Q.: Cross-modality bridging and knowledge transferring for image understanding. *IEEE Trans. Multimed.* **6**(3), 1–10 (2019)
4. Alvarez, L., Morales, F.: Affine morphological multiscale analysis of corners and multiple junctions. *Int. J. Comput. Vis.* **25**(2), 95–107 (1997)
5. Mokhtarian, F., Suomela, R.: Curvature scale space for robust image corner detection. In: 1998 IEEE International Conference on Pattern Recognition, pp. 1819–1823 (1998)
6. Alvarez, L.: Corner detection using the affine morphological scale space. In: 2017 IEEE International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2017), pp. 29–40 (2017)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference, pp. 147–151 (1988)
8. Rosten, E., Drummond, T.: FAST machine learning for High-speed corner detection. In: 2006 European Conference on Computer Vision, pp. 1–14 (2006)
9. Moravec, H.P.: Obstacle avoidance and navigation in the real world by a seeing Robot Rover. Technical Report, DTIC Document (1980)
10. Schmid, Cordelia, Mohr, Roger: Local gray value invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. Inst. Electr. Electron. Eng.* **19**(5), 530–534 (1997)
11. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
12. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: *Computer Vision Using Local Binary Patterns*, Computational Imaging and Vision Series, pp. 38–42. Springer, New York (2011)
13. Smith, S.M., Brady, J.M.: SUSAN—a new approach to low level image processing. Technical Report TR95SMS1c (patented), Crown Copyright (1995), Defence Research Agency, UK (1995)
14. Smith, S.M., Michael Brady, J.: SUSAN—a new approach to low level image processing. *Int. J. Comput. Vis. Arch.* **23**(1), 45–78 (1997)
15. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary Robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV 2011), pp. 1–8 (2011)
16. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision (ICCV 2011), pp. 1–8 (2011)
17. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: 11th European Conference on Computer Vision (ECCV 2010)
18. Calonder, Michael, et al.: BRIEF: computing a local binary descriptor very fast. *Pattern Anal. Mach. Intell.* **34**(7), 1281–1298 (2012)

19. Rosin, P.L.: Measuring corner properties. *Comput. Vis. Image Underst.* **73**(2), 291–307 (1999)
20. Davis, L.S.: A survey of edge detection techniques. *Comput. Graph. Image Process.* **4**(3), 248–260 (1975)
21. Dim, J.R., Takamura, T.: Alternative approach for satellite cloud classification: edge gradient application. *Adv. Meteorol.* **2013**, 1–8 (2013)
22. Hast, A.: Simple filter design for first and second order derivatives by a double filtering approach. *Pattern Recognit. Lett.* **42**(1), 65–71 (2014)
23. Gunn, S.R.: Edge detection error in the discrete Laplacian of Gaussian. In: 1998 International Conference on Image Processing (ICIP 98), pp. 515–519 (1998)
24. Krig, S.: *Computer Vision Metrics—Survey, Taxonomy, and Analysis*, pp. 370–378. Apress, New York (2014)
25. Rosten, E., Drummond, T.: Machine Learning for high-speed corner detection. In: 9th European Conference on Computer Vision, Graz, pp. 430–443 (2006)
26. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A.: A comparison of affine region detectors. *Int. J. Comput. Vis.* **65**(1–2), 43–72 (2005)
27. Balntas, V., Lenc, K., Vedaldi, A.: HPatches: a benchmark and evaluation of handcrafted and learned local descriptors. In: *Computer Vision & Pattern Recognition* (2017)

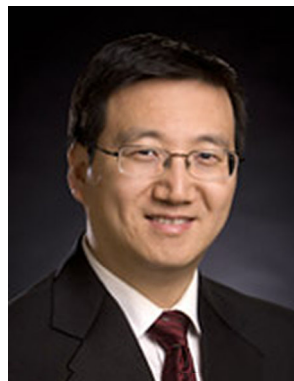
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Zhiyong Peng is a senior experimentalist in the College of Electronic Engineering and Automation at Guilin University of Electronic Technology (GUET), Guilin, China. He received his M.S. in Signal and Information Processing from GUET in 2006. He is currently pursuing the Ph.D. and studying in Oklahoma State University as a visiting scholar. His research interests include computer vision, photogrammetry and image processing.



Jun Wu is a Professor and Ph.D. supervisor in the College of Electronic Engineering and Automation at Guilin University of Electronic Technology, Guilin, China. He holds B.S. degrees and M.S. in Engineering both from Wuhan University, a Ph.D. in Photogrammetry and Remote Sensing from Wuhan University, and Postdoctoral research in Old Dominion University, USA. His research interests include robot vision, photogrammetry, in which areas he has published over 60 papers in both qualified journals and conferences. He is a corresponding author of this paper.



Guoliang Fan is a professor in the School of Electrical and Computer Engineering at Oklahoma State University (OSU). He is currently a specially appointed professor of the Shaanxi Hundred Talent program in School of Automation and information Engineering at Xi'an University of Technology, China. He received the B.S. degree in automation engineering from the Xi'an University of Technology, Xi'an, China, the M.S. degree in computer engineering from Xidian University, Xi'an, and the Ph.D. degree in electrical engineering from the University of Delaware, Newark, DE, USA, in 1993, 1996, and 2001, respectively. His current research interests include image processing, pattern recognition, and computer vision. He is an associate editor of *IEEE Transactions on Image Processing*, *IEEE Journal of Biomedical and Health Informatics* and *EURASIP Journal on Image and Video Processing*.