Journal of Computers Vol. 28, No. 5, 2017, pp. 255-275
doi:10.3966/199115992017102805023

Research on Visual Image Processing of Mobile Robot Based on OpenCV

Shu-Jing Zhang[1], Lei-Zhang[1*] and Ri Gao[1]

[1] School of Information and Electrical Engineering, Beijing University of Civil Engineering and Architecture
Beijing, China
zhanglei@bucea.edu.cn

Received 25 October 2016; Revised 13 June 2017; Accepted 26 June 2017

Abstract. This work takes the Mecanum wheel mobile robot as experimental platform, Microsoft Visual 2010 and OpenCV (Open Source Computer Vision Library) based on 2.4.9 platform for the development of research on mobile robot vision system. The main work is as follows: first, build the Mecanum wheel robot platform, realizes the mobile robot with full mobility. Second, the Kinect vision sensor is used to acquire the image information in the robot's visual field and to process the image. Including Canny edge detection, histogram equalization, Harris corner detection algorithm, sub-pixel level corner detection, localization, image affine transformation, Hof line transform, data processing center. Third, the obtained image is processed, and the image quality is improved by image preprocessing, so that the image clarity is improved. Based on the above image information processing, the algorithm is improved on the basis of Harris corner detection algorithm, and the corner coordinate is obtained at sub-pixel precision. Fourth, the processed image information is decided and analyzed in the database of the data center to form the motion control command of the mobile robot. The feasibility of the system is verified by experiments.

Keywords: image detection, image processing, machine vision, mobile robot, OpenCV

1 Introduction

Robot is a typical representative of mechatronic products. It is widely used in industry, agriculture, aeronautics and Astronautics, military and other fields. With the development of robotics, computer technology and image processing technology, the application of machine vision is more and more extensive. As the main way for robot to acquire environment information, machine vision can improve the ability of robot identification, localization and multi machine cooperation, and increase the flexibility of robot. Mobile robot acquires image information through vision, calculates geometric parameters and position of target, and is an important application of machine vision in robot technology [1].

Vision system is an important research field. Experts and scholars hope to simulate human vision by artificial means. With the development of the research, machine vision has been developed gradually, and the human visual function has been realized in the computer simulation environment. Machine vision uses video data and image data as raw data to study and extract and parse valid information in images and videos [2]. In the process of image acquisition, the camera usually contains all kinds of noise and interference, so it is necessary to process the image. Image processing can enhance the useful information of the image, improve the quality of the image, and facilitate computer analysis. The image processing process mainly includes the image pretreatment, the image segmentation, the image analysis and so on. Machine vision is widely used in the fields of manufacturing, product inspection and logistics. Machine vision can replace human vision in situations that are not suitable for manual work or where human vision is difficult to meet. At the same time, the use of machine vision can increase production efficiency and improve automation [3].

[*] Corresponding Author

Machine vision system can be roughly divided into two kinds according to their working environment. The first is the system known as the working environments, the second is the unknown working environment or the working environment with many uncertain factors. At present, in the field of machine vision, the working environment is well known and the environment is unknown or more uncertain factors. Further research is needed. The robot needs to complete a series of actions or specific behavior, obtaining environmental information by visual sensors, and then through the computer analysis of the visual information, the environment information related to the description, and to provide useful information for the robot [4]. Like human vision, the robot vision system will play an important guiding role in its behavior, enabling the robot to flexibly respond to changes in its surrounding environment. The robot perceives the information in the work environment or the information it is interested in through the visual system, and records it as a basis for decision making. Robot vision technology is mainly used in robot control are: to provide the description of the scene data, visual control system required target location or movement data, such as visual search and positioning system, visual tracking system etc.. This paper is to develop a robust control algorithm for a five-link biped robot with one degree of under actuation to climb stairs in a natural manner [5]. The paper use gesture recognition technology that utilizes the MS Kinect sensor to facilitate natural interactions between humans and robots. To evaluate gesture recognition performance in a real-life environment, they constructed a new gesture database that takes into account cluttered backgrounds [6].

2 Mobile Robot Platform

2.1 Mecanum Wheel Mechanical Structure

The selection of mobile robot platform needs to consider many aspects, including the size of mobile robot itself, the materials used, and the overall beauty. When the weight of the mobile robot exceeds a certain value, the material of the wheel is required. Wheels should have good rigidity and elasticity, and sometimes use inflatable materials. The deformation does not be too large, and the friction force should be increased or decreased according to the needs of the task. Considering all factors, the mobile platform chassis wheel structure will install the Mecanum wheel with comprehensive performance of the whole direction of travel [7].

From the perspective of robot kinematics, Mecanum wheel to realize the full range of mobile, the Jacobian matrix equation must be of full rank. Control system required from the mobile robot and drive of this point, when the Mecanum wheel is driving wheel, the mobile robot must remain in three degrees of freedom. Only satisfy these two angle condition, can really attain its Omni-directional comprehensive movement request [8].

In the Fig. 1 and Fig. 2, the digital symbol is the Mecanum wheel. The inclined part of the upper part is the angle between the axis of the roller and the steel ring. The geometric position and symmetry point of the shape of the device is the spatial origin of the coordinate axis of the motion mode. It can be concluded that the above six forms of structural distribution conform to such conditions: first, the axis of the roller is identical with the angle of the wheel rim. Second, each of the two wheels is symmetrical with each other. Third, the center position of the platform chassis is symmetrical.

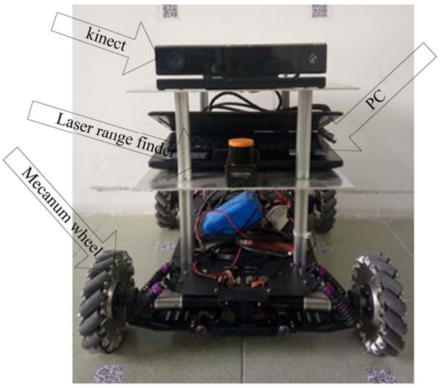


Fig. 1. Mobile robot model

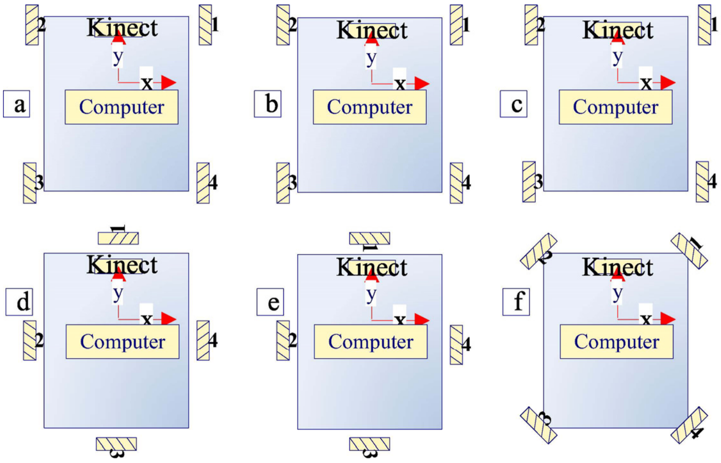


Fig. 2. Mobile chassis layout

The above six kinds of layout, (a), (b) and (d) Jacobi matrix equation three case is full rank, to meet the initial conditions of robot kinematics. (c), (e) and (f) the three cases can directly determine the function without omnidirectional movement. Further, (a), (b) and (d) three cases, (a) the angle between the roller and the Mecanum wheel rims are tilted 45^0, the layout is square. In this case, the speed of each wheel is the same as that of the mobile robot in general, so that there is no rotation around the axis. (d) the condition leads to only two degrees of freedom in the platform movement, and therefore does not have the capability of All-Round comprehensive movement. On the whole, the best layout is the (b) model [9].

(1) The occurrence of such situations will Mecanum wheel in the whole system began marching:

(2) The wheel integrally rotates along its own central axis.

Mecanum wheel on the roller will hinder the force along the barrel axis gyro.

Mobile robots Four Mecanum rounds, each McNeil wheel will be connected to a motor by a gear unit. That is, each wheel will have its own independent direction of operation. The four different combinations of Mecanum wheels, different running directions, will eventually make mobile robots produce different modes of movement.

The motor makes the Mecanum wheel drive shaft to rotate along the vertical direction of the motor, no power roller for rotating Mecanum wheel and passive operation. A Mecanum wheel alone in these two methods will have a final force. By controlling the motor speed and direction, the four Mecanum wheel will produce a combination of speed, and the speed can be an arbitrary direction. In this way, the mobile platform has the ability of comprehensive and comprehensive movement [10]. As shown in Table 1, for the Mecanum wheel in the rotation direction under different operation mode corresponding to the mobile robot. W1, W2, W3 and W4 is four Mecanum wheel. “+” Mecanum wheel forward rotation, “-” Mecanum wheel reverse rotation, “*” means no rotation.

Table 1. Mobile robot movement

	forward	reverse	right straight	left straight	right anterior 45^0	left anterior45^0	clockwise rotation	Anticlockwise rotation
W1	+	-	-	+	*	+	-	+
W2	+	-	+	-	+	*	+	-
W3	+	-	-	+	*	+	-	+
W4	+	-	+	-	+	*	+	-

As shown in Fig. 3, a schematic diagram of the platform multi-mode movement mode is given.

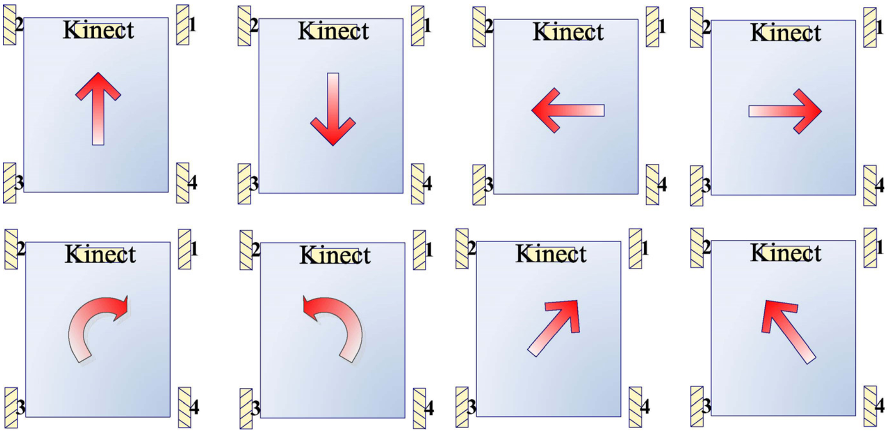


Fig. 3. Robot multi-mode movement mode

2.2 Mobile Robot Motion Control System Based on CAN Communication

This system is mainly composed of the host computer, the mobile robot server and database, through wireless LAN communication between the three upper machine is a PC machine, is responsible for sensor information data; machine vision sensor, including driving module, power supply module of the mobile robot, and the mobile platform Mecanum wheel device, is responsible for the implementation of robot motion control system.

As shown in Fig. 4, the overall structure of the motion control system for the mobile robot is shown. It is divided into five parts, including power module, lower computer, communication module, driver and motor. Among them, the small computer is used as the bottom part of the lower computer. After the lower computer scans and receives the valid instructions in the database, the driver can be controlled by the CAN bus communication module. Drive part, this paper uses different drives according to the difference of mobile platform. After the signal is received, the driver will control the motor operation, so as to realize the overall motion control of the platform. The power module is responsible for powering the lower computer and the drive unit [11].

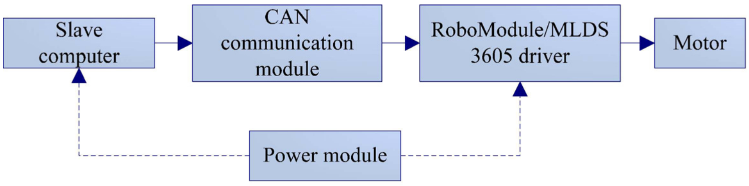


Fig. 4. Overall structure of control system

3 Visual Image Processing

Visual image processing is the use of visual sensors instead of visual organs as input devices, by computer instead of the brain to complete the processing and analysis. The ultimate goal of visual image processing is the computer like people to observe and understand the visual world, with independent ability to adapt to the environment.

3.1 Image Preprocessing

In the machine vision system, visual information processing technology mainly depends on the method of image processing, including image transform, image compression encoding, storage, preprocessing, image enhancement and restoration, image segmentation, feature extraction, image description, image classification, model matching and so on. After image processing, the image quality is improved, so you can easily extract some features of the images contain or special information, but also through the image data conversion, encoding and compression, can facilitate the image storage and transmission. The robot vision system is oriented to the 3D scene, but most of them are still aimed at the two-dimensional image processing. Normally, two-dimensional images obtained directly from image acquisition devices contain a wide variety of random noises and distortions. In order to enhance the detect ability and minimize data relevant information, so as to improve the reliability of feature extraction, image segmentation, matching and recognition, in order to make the correct action plan then, need to remove noise, distortion and correction of the original image [12]. Thus, the useful information is suppressed, the useless information is suppressed, and the quality of the image is improved.

In the process of image processing, the point is not to extract the information from the image, but to remove the wrong and not much valuable information. Because of the different image sampling environment, such as the sensitivity of illumination and the performance of the device, there may be some defects such as noise and lack of contrast. In addition, the size and location of the target in the image are uncertain because of the different distance and focal distance. In order to ensure the consistency of the target size, location and target image quality, it is necessary to preprocess the image. After pretreatment, the quality of the output image is improved to a certain extent, and the visual effect of the image can be improved, so that the robot can deal with image analysis, processing, understanding and recognition.

The image pre-processing technology used here is mainly Canny edge detection algorithm processing technology. Because of the change of light, there is a bright change of environment image. In order to decrease the color deviation of light, the ray compensation algorithm is added in image preprocessing. The corresponding ray compensation coefficients are calculated mainly by calculating the variance of the gray values of all pixels in the image. Finally, some compensation systems are multiplied by the corresponding pixel RGB component. The gray value Gray of its pixel point can be converted by the following formula:

$$Gray = (R*299 + G*587 + B*114)/100 \tag{1}$$

The number of pixels of different gray value statistics, then meet the critical condition of gray level and pixel gray value is obtained according to the proportion of the total coefficient set, calculate the average gray value of the image, with the average gray level 255 is divided by the light compensation coefficient, finally all the RGB components according to the light compensation amplification coefficient, a numeric value greater than 255 set 255, to adjust the RGB value of image [13].

3.2 Canny Edge Detection

The first step of edge detection image segmentation is to identify the points of brightness change in digital images. Edges are considered as places where a certain amount of illumination varies, and there is always an edge between two adjacent regions with different gray values. In essence, the edge of an image is a reflection of the local characteristics of a small continuity, for example, gray mutation, color mutation, texture changes, etc., which marks the end of an area and the beginning of another area. This discontinuity can be easily detected by derivative, which is usually detected by first and two order derivatives. In fact, the derivative in digital image is differentiated by differential approximation. The classical method of edge extraction is to investigate the change of gray level in the neighborhood of each pixel of an image. The edge feature is extracted by changing the first or two order derivative of the neighborhood, that is, the edge detection by operator method. This paper uses the Canny operator of edge detection, after preprocessing, image quality is improved to a certain extent, it can improve the visual effect of the image or more convenient for the robot to image analysis and image processing, understanding and recognition.

Canny detection criteria.

(1)Good edge detection performance: that is, as little as possible the real edge points as non-edge points, and at the same time as little as possible non edge points as edge points.

(2)Accurate edge point: that is, if the edge of the image is not the width of a pixel, then select the location of the edge point, as far as possible in the center of the real edge of the image.

(3)Single edge response: if the edge of the image is more than one pixel width, then each edge has and only one point is retained, so that the edges are preserved to the maximum extent.

The above is the Canny edge detection criterion, and the Canny three criterion is described in mathematical expressions. In this way, the mathematical expression of the optimal edge detection operator can be deduced from the mathematical point of view, and finally the optimal edge detection operator is obtained by some optimization and approximation [14].

The So-called good edge detection performance is to minimize the actual edge points as non-edge points, but also as little as possible to non-edge points as edge points. The signal detection theory, with the increase of the SNR of image can satisfy these two aspects as little as possible, so the first rule of Canny three standards is for a f(x) function, this function to make the image detection after edge point noise most larger than SNR. After f(x) filtering, the response of the edge point is:

$$H_G = \int_{-w}^{w} G(-x) f(x) dx \quad \textbf{(2)}$$

The square root response of the noise is

$$H_n = n_0 \sqrt{\int_{-w}^{w} f^2(x) dx} \quad \textbf{(3)}$$

n_0 is the mean square deviation of noise.

Therefore, the mathematical expression of the first criterion of Canny can be obtained from the above derivation:

$$SNR = \frac{\left| \int_{-w}^{w} G(-x) f(x) dx \right|}{n_0 \sqrt{\int_{-w}^{w} f^2(x) dx}} \quad \textbf{(4)}$$

Accurate edge localization. Assuming that the edge position detected by the algorithm is at $x = x_0$, the $H_G(x) + H_n(x)$ takes the maximum value at x_0, therefore:

$$H_G(x) + H_n(x_0) = 0 \quad \textbf{(5)}$$

Since the center position of the $G(x)$ is assumed to be at $x = 0$, that is, the actual edge position is at $x = 0$, the $H_G(x_0)$ makes a maximum at $x_0 = 0$, therefore: C.

Using the Taylor formula, the $H_G(x_0)$ is expanded at $x_0 = 0$:

$$H_G(x_0) = H_G(0) + H_G^{'}(0) x_0 + O(x_0^2) \quad \textbf{(6)}$$

Formula 5 to formula 6, can get the approximate value of $H_G(x_0)$ for:

$$H_G(x_0) \approx H_G^{''}(0) x_0 \quad \textbf{(7)}$$

Formula 7 into the formula 4 again:

$$H_G^{''}(0) x_0 \approx -H_n(x_0) \quad \textbf{(8)}$$

Because

$$E\left[H_n^{'}(x_0)^2\right] n_0^2 \int_{-w}^{w} f'^2(x) dx \quad \textbf{(9)}$$

Where $E(x)$ stands for X expectations, so:

$$E(x_0^2) \approx \frac{n_0^2 \int_{-w}^{w} f'^2(x) dx}{\left[\int_{-w}^{w} G(-x) f'(x) dx \right]^2} \quad \textbf{(10)}$$

The mathematical expressions for the second location criteria are defined as:

$$Localizati\ on = \frac{\left| \int_{-w}^{w} G(-x) f'(x) dx \right|}{n_0 \sqrt{\int_{-w}^{w} f'^2(x) dx}} \quad \textbf{(11)}$$

Among them, $f'(x)$ and $G(-x)$ are derivatives of $f(x)$ and $G(-x)$ respectively. Since the smaller the $E(x^2)$, the more accurate the edge of the image is located, so the search for a filter function $f(x)$ makes it possible for $f(x)$ to satisfy the maximum value obtained by localization [15].

Combining the first criterion, the J of the filter function $f(x)$ is obtained and the maximum value is obtained. The expression for J is:

$$J = \frac{\left|\int_{-w}^{w} G(-x) f(x) dx\right|}{n_0 \sqrt{\int_{-w}^{w} f^2(x) dx}} \frac{\left|\int_{-w}^{w} G(-x) f'(x) dx\right|}{n_0 \sqrt{\int_{-w}^{w} f'^2(x) dx}} \tag{12}$$

As shown in the following Fig. 5. Fig. 5 is the Canny algoritm flow chart. Fig. 6 and Fig. 7 are the result of corridor and indoor scene Canny algorithm image detection.

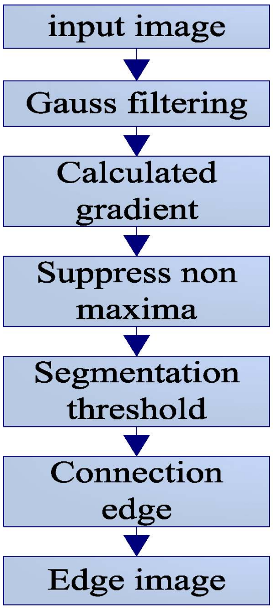


Fig. 5. Canny algorithm flow chart

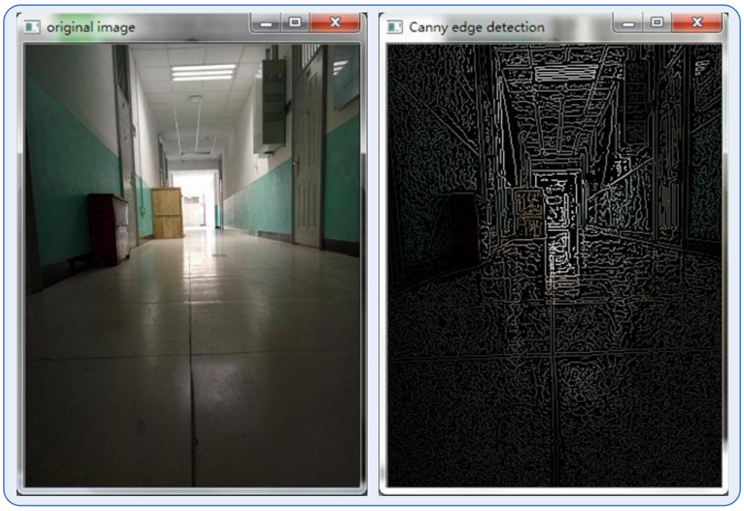


Fig. 6. Corridor scene Canny algorithm detection

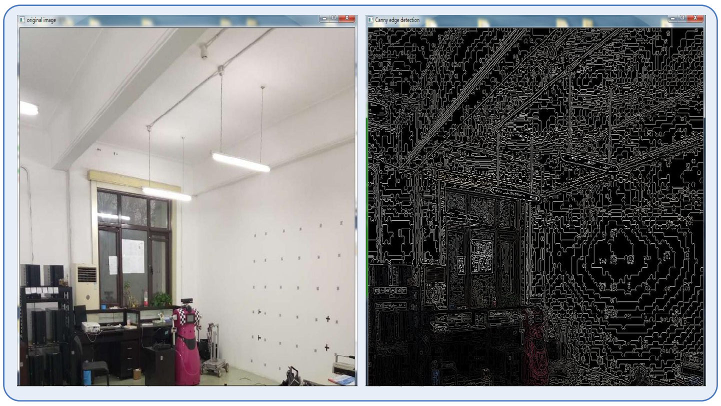


Fig. 7. Indoor scene Canny algorithm detection

3.3 Histogram Equalization

Histogram equalization enables the histogram of the original image to be evenly distributed over the whole gray range, and enhances the contrast of the whole. Histogram equalization is represented by formula 13:

$$p(s_k) = \frac{n_k}{n}, k = 0, 1, 2 \ldots 255 \quad \textbf{(13)}$$

n represents the total number of pixels in the image, and $p(s_k)$ represents the probability of the occurrence of gray level K of the image $f(x, y)$. It is the gray value of level K gray, and n_k represents the number of pixels in the image whose gray value is s_k.

Histogram equalization is an important application of gray level transformation, which is mainly used in image enhancement processing. The pixel gray level change of the image is random. When the histogram of the image in the robot vision field is uneven and the brightness is uneven, the histogram equalization algorithm can make the histogram approximately flat. The equalized image can only be approximately uniform distribution [16]. The essence of dynamic range image equalization expansion is to expand the quantization interval, but the quantization level decreased, therefore, the original pixels of different gray levels after treatment may be the same, the same form a gray area, with clear boundaries between regions, and the emergence of false contour from.

If the original image contrast is high, if the equalization is gray, the contrast is reduced. In the image of white ease, equalization will merge some pixel, thereby increasing the contrast. If the equalized image is equalized, the image will not change. Histogram equalization algorithm is to each gray level histogram was normalized for the cumulative distribution of each gray level, a gray mapping table, then according to the correction of each pixel in the original image corresponding to the gray value.

The classic histogram equalization algorithm may have the following shortcomings:

The gray range of the output image is difficult to reach the maximum gray range permitted by the image format.

Although the distribution histogram of the output image is close to the uniform distribution, there is still a big difference between its value and the ideal value 1/n, which is not the best value.

The gray level of the output image may be merged too much. Because of the phagocytosis of gray level, it is easy to cause the loss of image information.

The histogram algorithm is improved, so the result of image processing is shown in the Fig. 8 and Fig. 9.

Probability density function [17]. In order to facilitate computation, we need to normalize the histogram, that is, the gray range from 0~255 to 0~1. Normalized histogram is actually a probability density function (PDF, probability, density, function), equalization is to make the probability density of 1.

The use of Pr (R) to represent the original image PDF, Ps (s) said after the equalization of PDF, R and s respectively represent the gray values before and after equalization, $r, s \in \lfloor 0,1 \rfloor$. According to the theory of probability, we can draw:

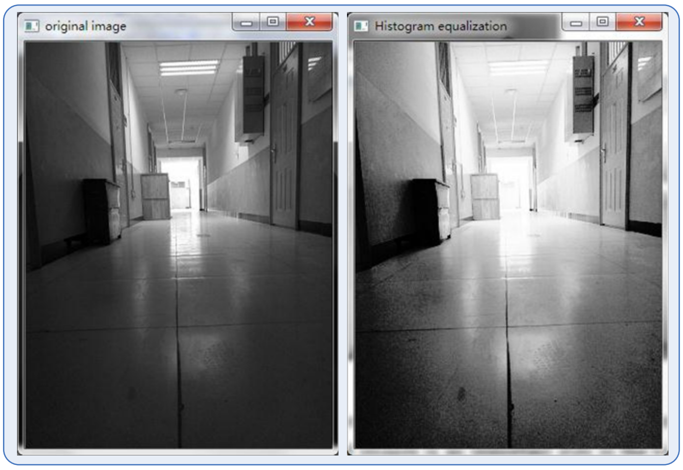


Fig. 8. Corridor scene histogram equalization

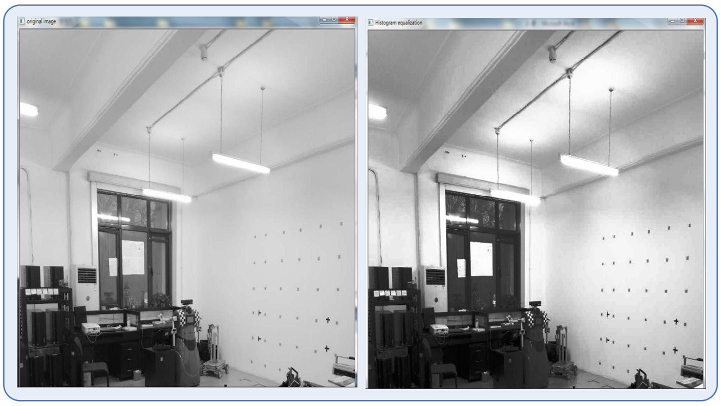


Fig. 9. Indoor scene histogram equalization

$$P_s(s)=\left[\frac{P_r(r)}{\left|\frac{ds}{dr}\right|}\right]_{r=T^{-1}(s)} \tag{14}$$

In the formula, $T^{-1}(s)$ represents the inverse transformation function of T (r).

$P_s(s)=\frac{P_r(r)}{P_r(r)}=1$, the probability density is 1, or $\frac{ds}{dr}=P_r(r)$, $ds=P_r(r)\times dr$.

The R integral on both sides of the equation obtains the equilibrium formula of PDF: $s=T[r]=\int_0^r P_r(w)dw$

In the formula, T (r) represents the grey scale transformation function of R, $\int$ means integral, and W is assumed variable.

Cumulative distribution function

For the image, the discrete form of the formula is used, and the probability of the occurrence of a gray level pixel is: $P_r(r_k)=n_k/N$

$P_r(r_k)$ is the probability of the K binary image pixel gray level appearance, rk is the first k gray level, the current level of k, $k\in\lfloor 0,1\rfloor$. nk is the number of rk pixels, and N is the total number of pixels in the image. Histogram equalization formula of image:

$$s = T(r_k) = \sum_{j=0}^{k} n_j \big/ N = \sum_{j=0}^{k} P_r(r_k) \quad \textbf{(15)}$$

In the formula, T (rk) is used to represent the K gray level conversion function of the original image. A represents the ratio of the total number of pixels and the sum of the number of pixels of 0~j grayscale, it is said in front of the percentile (all pixels number / total number of pixels in the color and color front). The $\sum P_r(r_k)$ represents the grey level occurrence probability of 0~k. Because s is the normalized value $s \in \lfloor 0,1 \rfloor$, to be converted to 0~255 color value, need to be multiplied by 255, namely $s = \sum P_r(r_k) * 255$. This transformation formula is also called the cumulative distribution function of the image.

3.4 Harris Corner Detection Algorithm

Corner points contain much important image information, and they are important features of images. Extracting the corner points of images is an important step in the vision system of mobile robots. There are many kinds of corner expressions, such as in the image boundary curvature made great point; image boundary curvature changes; discontinuous changes in image edge gradient direction; the image gradient ratio and gradient value are great points and so on [18]. The definition and description of corner points mainly include the following:

(1)The corner is the intersection point of two or more than two straight lines or curves;

(2)The corner indicates the direction of the object edge change;

(3)The corner is the gradient and gradient change rate of the largest point in the image.

Generally speaking, the corner points and the surrounding points are very different. It represents the position of the gray value of the image in the two-dimensional space.

Harris algorithm is computationally simple and convenient, and has high detection efficiency. The corner points are detected by curvature and gradient of corner points, and other features of image are less dependent. Moravec corner detection algorithm is the basis of Harris corner detection algorithm. It proposes an idea based on local region feature. In 1977, Moravec proposed the Moravec operator, which determined the corner points by thresholding and Non-maximum suppression on the basis that the gray values of each corner varied greatly along the corners. The Moravec operator is calculated only in the discrete direction, the difference between the Harris corner detection algorithm and Morvaec corner detection for Harris operator allows to obtain the gray change in all directions, namely, the Harris operator can calculate the gray change, all directions so that a wider range of Harris operator [19]. Similar to the Moravec operator to calculate the gray change in a certain direction, we use (x_0, y_0) as the center of a circle to calculate, remember the circle $0_0 = N(x_0, y_0, r)$, gray function $I(x, y)$ at (x_0, y_0), along a certain direction of the gray change, such as formula 16:

$$f(x_0, y_0) = \sum_{(x,y)\in O_0} [I(x+\Delta x, y+\Delta y) - I(x, y)]^2, (\Delta x, \Delta y) \in D_\varepsilon \quad \textbf{(16)}$$

In formula 16, (x, y) is the pixel coordinate, $I(x, y)$ is the gray value of the point, $N(x_0, y_0, r) = \{(x, y) \in R^2, \|(x, y) - (x_0, y_0)\| \le r\}$ is a neighborhood of (x_0, y_0), and $(\Delta x, \Delta y)$ is the translation unit, $(\Delta x, \Delta y) \in R^2$, $D_\varepsilon = \{(\Delta x_0, \Delta y_0 \in R^2), \|(\Delta x_0, \Delta y_0)\| = \varepsilon, \varepsilon > 0\}$

$$\nabla I(x, y) = (\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y})^T \quad \textbf{(17)}$$

Remember the formula 17 gradient (x, y), 16 $I(x+\Delta x, y+\Delta y)$ will use Taylor formula and ignore the infinite unfolding events, represented by the former two approximation can be:

$$I(x+\Delta x, y+\Delta y) \approx I(x, y) + \nabla(x, y)^T \begin{bmatrix} \nabla x \\ \nabla y \end{bmatrix} \quad \textbf{(18)}$$

Substituting formula 18 into formula 16 can be obtained:

$$f(x_0, y_0) \approx \sum_{(x,y)\in N_0} \left[\Delta x, \Delta y\right] \nabla I(x,y) \nabla I(x,y)^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, (\Delta x, \Delta y) \in D_\varepsilon \tag{19}$$

Transform formula 19:

$$f(x_0, y_0) \approx \left[\Delta x, \Delta y\right] \sum_{(x,y)\in N_0} (\nabla I(x,y) \nabla I(x,y))^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = [\Delta x, \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, (\Delta x, \Delta y) \in D_\varepsilon \tag{20}$$

$M = \sum_{(x,y)\in N_0} \nabla I(x,y) \nabla I(x,y)^T = (\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y})^T (\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y})$, is a symmetric semi positive definite matrix.

In order to filter the noise better, Gauss window is used to filter the window function:

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \tag{21}$$

From formula 20 and formula 21, the image gray autocorrelation function is introduced:

$$E(x,y) = \sum_{u,v} G_{(u,v)} [I(x+u, x+y) - I(u,v)]^2 \tag{22}$$

The formula 22, u, v Gauss said the size of the filter window.

By formula 17, transform 22:

$$E(x,y) = Ax^2 + 2Cxy + By^2 = (x,y) \begin{bmatrix} A & C \\ C & B \end{bmatrix} (x,y)^T \tag{23}$$

Where $A = (\frac{\partial I(X,Y)}{\partial X})^2 \otimes G,\quad B = (\frac{\partial I(X,Y)}{\partial Y})^2 \otimes G,\quad C = (\frac{\partial I(X,Y)}{\partial X})(\frac{\partial I(X,Y)}{\partial Y}) \otimes G,\quad \otimes$ denotes convolution.

The matrix M is the approximate Hessian matrix of the autocorrelation function $E(x,y)$,

$$M(X,Y) = \begin{bmatrix} A(X,Y) & C(X,Y) \\ C(X,Y) & B(X,Y) \end{bmatrix} \tag{24}$$

The eigenvalues of matrix M are used to approximate the curvature extremum of the image gray correlation function, so that λ_1 and λ_2 are the eigenvalues of matrix M, and the values of λ in three cases are considered [20].

The two eigenvalues tend to be zero. When the two eigenvalues are approximately zero, the gray change in the window region of the image is very small, indicating that the center point is in the interior of the region, not the corner point.

In the two eigenvalue, one eigenvalue is much smaller than the other eigenvalue. When one eigenvalue is much smaller than the other one, the gray level in the image window region varies greatly in one direction, and the other direction is very small, indicating that the center point is on the edge, not the corner point.

Two eigenvalue values are great. When the values of the two eigenvalues are large, the gray regions of the moving image window region in any direction are very large, which is the corner point.

To simplify the calculation, the response function is shown in equation 25, by defining the response function to avoid the solution of eigenvalues:

$$R(X,Y) = \det(M) - k(trace(M))^2 \tag{25}$$

Among them, $\det(M) = \lambda_1\lambda_2 = AB - C^2,\quad trace(M) = \lambda_1 + \lambda_2 = A + B,$ K is constant, according to experience, generally set k = 0.04.

When the $R(X,Y)$ value is minimum and approximately zero, then the judgment point is in the region.

When the $R(X,Y)$ is greater than the threshold, the point is the corner point. When the $R(X,Y)$ value is less than zero, the decision point is the edge point. The Fig. 10 and Fig. 11 as follows are shown the corridor scene and indoor scene harris corner detection.

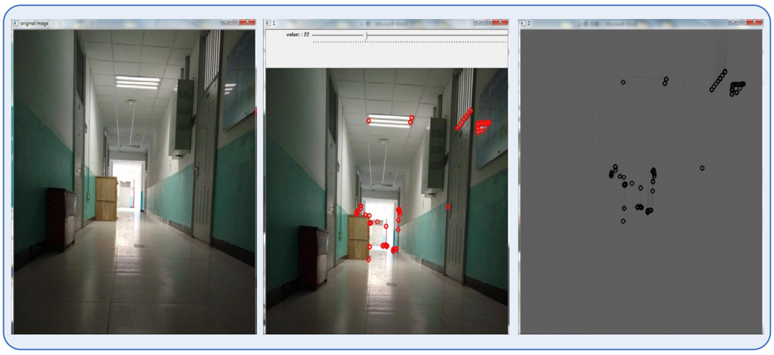


Fig. 10. Corridor scene harris corner detection

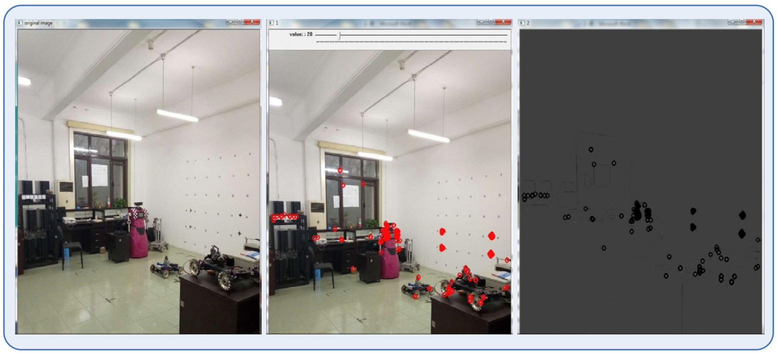


Fig. 11. Indoor scene harris corner detection

3.5 Sub-pixel Corner Detection and Location

Based on the Harris corner detection, Harris corner detection and sub-pixel corner detection by combining the initial coordinates of corner points using Harris corner detection value, on the basis of sub-pixel corner detection.

The core of Sub-pixel edge location algorithm based on interpolation is the gray level of the pixel values or derivative gray value interpolation, the increase of information, in order to achieve sub-pixel edge detection, it is the first step, on the edge of the coarse positioning with the traditional edge detection operator, then using the two order derivative edge points zero this characteristic, 3 points in a small neighborhood of the edge point on both sides, if too often leads to inaccurate positioning, using gradient several points value interpolation, interpolation function in solving the unknown parameters, the maximum point for the final function, namely zero a derivative, then the sub-pixel position of this point is that we want the edge [21].

Based on sub-pixel edge location algorithm of interpolation in the study is to interpolate the gray value of the first derivative or gradient method two times interpolation, spline interpolation and Chebyshev polynomial interpolation. The interpolation class has a short operation time, and the two interpolation algorithm is simple and can be implemented by hardware, which is suitable for on-line detection. When the linear diffusion function of optical system is symmetrical, the interpolation edge detection has higher precision. However, for noise sensitive, if the rough localization of edges is wrong due to noise, then the sub-pixel location of the edge is also incorrect.

The core of sub-pixel edge location algorithm based on moment preserving is: using the pixel gray value and gray distribution model including edge position, calculation of various statistics, such as the gray moment, spatial moment and centroid value and the local energy value of sub-pixel edge detection. Based on the method of moments, the analytic closed solutions of edge positions can be obtained, and the detection time is short and the edge detection accuracy is higher.

The kernel of the sub-pixel edge location algorithm based on fitting is to study the gray distribution of object edges and detect the edge position. The ability of fitting anti image noise is strong, and the edge detection precision is high. The detection precision of the ideal edge image in synthetic image or laboratory can reach one point of one pixel [22].

Sub-pixel positioning is to calculate the real location of the image in the feature, while the real position is sometimes not in the coordinates of the integer position of the pixel, but in the interior of the pixel. Sub-pixel positioning mainly has the following factors:

(1)Digital images are usually discretized into pixel forms;

(2)Each pixel corresponds to an integer coordinate position;

(3)Integer coordinates are not accurate for many applications, such as tracking, camera calibration, image registration, image stitching, and 3D reconstruction;

(4)To achieve the accuracy of some applications, you need to be accurate to the floating-point coordinate position.

The use of commonly used algorithms for positioning, initial positioning accuracy generally reached the integer coordinate location, usually at the beginning of the positioning accuracy to ensure that at least in a position near or far off, algorithm may also be effective; iterative matching model for accurate positioning, image feature detection feature location, the position precision reach the sub-pixel level. The purpose is usually iterative parameter optimization model.

Point positioning: a “point” usually takes more than one pixel, and does not have strong edge properties, and its edges are usually smooth and fuzzy. The Fig. 12 is a pixel point. It is not in a pixel region as a “point” position, should be based on a coordinate point to mark the “point” position, usually calculated area center position or area of the brightest coordinates of the corresponding position, using the Gauss model [23].

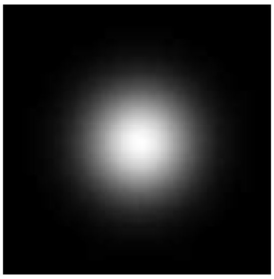


Fig. 12. Pixel point

$g(x, y) = \exp(-\frac{x^2 + y^2}{2\sigma^2})$, the final model function M is constructed using the Gauss model, as follows:

$$M(x, y, A, B, \sigma, u, v) = A + B\exp(\frac{(x-u)^2 + (y-v)^2}{2\sigma^2}) \quad \textbf{(26)}$$

M: The intensity of corresponding pixel location; (x, y): a pixel position in the image; A: strong background; B: bright peak in the region is strong; (U, V): bright peak location area; σ: Gauss model variance. On the left is $M(X, \theta)$, which simplifies the expression of $X = (X, Y)^T$, said the pixel position of the image; $\theta = (A, B, \sigma, u, v)^T$ is the parameters of the model points, when the model parameters are known, then the image feature has been located, namely the center position of the (*u*, *v*) Gauss model, the coordinate position is floating-point, namely the sub-pixel.

In view of the existing situation of “point” in the image, if the model is $M(X, \theta) = I(X)$, which corresponds to the strength and function model of M pixel position estimate of the same intensity, pixel matching point area then use the value of this model is correct, the model is suitable for describing the characteristics of the region. In fact, in some areas is through residual analysis, such as: $E(\theta) = \sum_{x \in W} [M(X, \theta) - I(X)]^2$. W, which is equivalent to a window or a template in the residual calculation before the first use of a simple algorithm for coarse positioning of a position, will be moved to the center of the window position, then the residual calculation is to minimize the error of $E(\theta)$, you can get the sub-pixel coordinates (U, V). The sub-pixel location method can directly calculate the derivative function of each parameter of the error function, or we can also use gradient descent method to solve the problem, and finally obtain the related parameters of the model [24].

The intersection point of an edge, and the angle is perpendicular to the gradient of the boundary line and the edge point.

As shown above Fig. 13, assume that a starting angle q is near the actual sub-pixel corner. The p point is in the neighborhood near the q point. If the P point is inside the uniform region, the gradient of the p point is 0; if the p point is on the edge, then the gradient direction of the p point is perpendicular to the edge direction. If the vector q-p direction is the same as the edge direction, then the gradient vector dot product of the q-p vector and the p point is 0. We can collect a lot of gradient group point in the vicinity of the initial corner and vector q-p the q that we needed more accurate corner position, then the vector point of each product group is set to 0, it is based on this idea, the dot product of 0 equations are combined to form a system of equations, the system equation is more accurate sub-pixel corners [25].

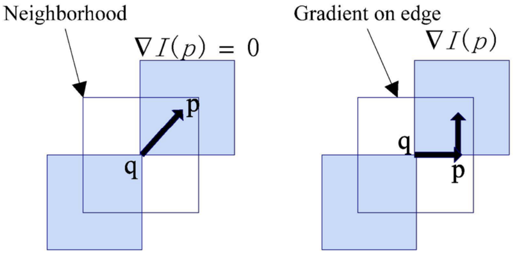


Fig. 13. a. p in a region   b. p on a region

Based on the above method, it is proved that the sub-pixel position information can be obtained with high precision. The iterative algorithm is adopted to obtain the sub-pixel position of the corner.

The implementation of sub-pixel corner location is based on the observation of vector orthogonality, that is, the vectors from the central point Q to its field point P and the gradient of the image at the P point are orthogonal, and the following expressions are obtained:

$$\varepsilon_j = DI_{pi}^T \cdot (q - p_i) \tag{27}$$

Where DI_{pi} represents the image gradient at P_i at a neighborhood point of q, based on the above method, the value of q is obtained by minimizing εi. When the εi is set to 0, the two sides are multiplied by DI_{pi}, and an equation about the neighborhood point can be obtained. The transformation of all neighborhood points, the equality of the left and right sides can be summed up, can be the following equation [26]:

$$\sum_i (DI_{pi} \cdot DI_{pi}^T) q - \sum_i (DI_{pi} \cdot DI_{pi}^T \cdot p_i) = 0 \tag{28}$$

Set $G = \sum_i (DI_{pi} \cdot DI_{pi}^T), b = \sum_i (DI_{pi} \cdot DI_{pi} \cdot DI_{pi}^T \cdot p_i) = 0,$ and get $q = G^1 \cdot b.$

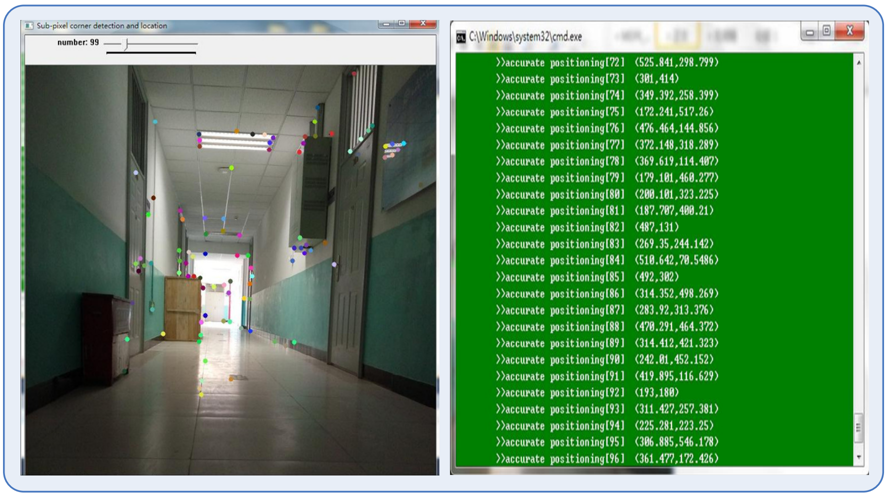
Sub-pixel corner location algorithm step as follows, the Fig. 14 and Fig. 15 showing the precise location of different scenes:

(1)Use the initial corner position and window size to calculate the position information and gray level information of all the points in the neighborhood of the window neighborhood;

(2)According to the location information of neighborhood points in the image, the gradient information of neighborhood points is calculated;

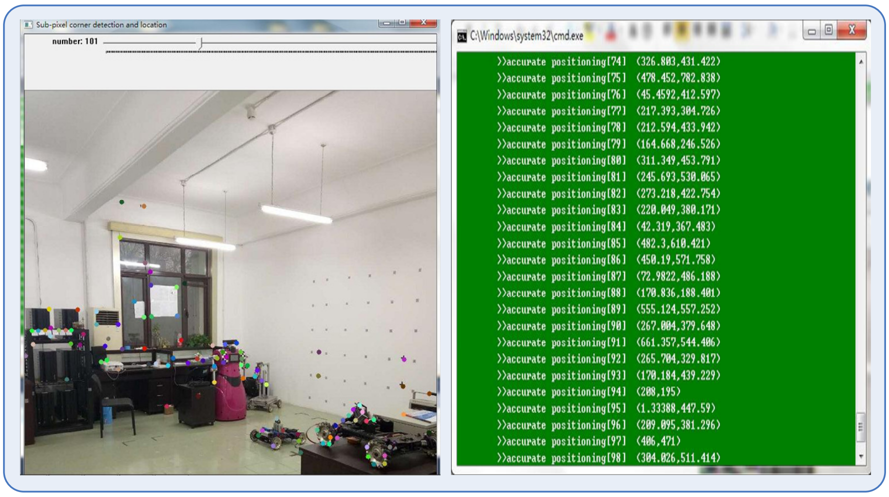
(3)Using the location information of the neighborhood point (this position information is relative to the window) and gradient information, we calculate the more accurate location of the corner by combining the above formula;

(4)Repeated iterations, until the end of the iteration is satisfied, the q is the sub-pixel position of the final corner.


Sub-pixel corner detection and location
number: 99
C:\Windows\system32\cmd.exe

```
>>accurate positioning[72]  (525.841,298.799)
>>accurate positioning[73]  (301,414)
>>accurate positioning[74]  (349.392,258.399)
>>accurate positioning[75]  (172.241,517.26)
>>accurate positioning[76]  (476.464,144.856)
>>accurate positioning[77]  (372.148,318.289)
>>accurate positioning[78]  (369.619,114.407)
>>accurate positioning[79]  (179.101,460.277)
>>accurate positioning[80]  (200.101,323.225)
>>accurate positioning[81]  (187.707,400.21)
>>accurate positioning[82]  (487,131)
>>accurate positioning[83]  (269.35,244.142)
>>accurate positioning[84]  (510.642,70.5486)
>>accurate positioning[85]  (492,302)
>>accurate positioning[86]  (314.352,498.269)
>>accurate positioning[87]  (283.92,313.376)
>>accurate positioning[88]  (470.291,464.372)
>>accurate positioning[89]  (314.412,421.323)
>>accurate positioning[90]  (242.01,452.152)
>>accurate positioning[91]  (419.895,116.629)
>>accurate positioning[92]  (193,180)
>>accurate positioning[93]  (311.427,257.381)
>>accurate positioning[94]  (225.281,223.25)
>>accurate positioning[95]  (306.885,546.178)
>>accurate positioning[96]  (361.477,172.426)
```

Fig. 14. Corridor scene sub-pixel positioning


Sub-pixel corner detection and location
number: 101
C:\Windows\system32\cmd.exe

```
>>accurate positioning[74]  (326.803,431.422)
>>accurate positioning[75]  (478.452,782.838)
>>accurate positioning[76]  (45.4592,412.597)
>>accurate positioning[77]  (217.393,304.726)
>>accurate positioning[78]  (212.594,433.942)
>>accurate positioning[79]  (164.668,246.526)
>>accurate positioning[80]  (311.349,453.791)
>>accurate positioning[81]  (245.693,530.065)
>>accurate positioning[82]  (273.218,422.754)
>>accurate positioning[83]  (220.049,380.171)
>>accurate positioning[84]  (42.319,367.483)
>>accurate positioning[85]  (482.3,610.421)
>>accurate positioning[86]  (450.19,571.758)
>>accurate positioning[87]  (72.9822,486.188)
>>accurate positioning[88]  (170.836,188.401)
>>accurate positioning[89]  (555.124,557.252)
>>accurate positioning[90]  (267.004,379.648)
>>accurate positioning[91]  (661.357,544.406)
>>accurate positioning[92]  (265.704,329.817)
>>accurate positioning[93]  (170.184,439.229)
>>accurate positioning[94]  (208,195)
>>accurate positioning[95]  (1.33388,447.59)
>>accurate positioning[96]  (209.095,381.296)
>>accurate positioning[97]  (406,471)
>>accurate positioning[98]  (304.026,511.414)
```

Fig. 15. Indoor scene sub-pixel positioning

3.6 Affine Transformation of Image

Affine transformation, also called affine mapping, refers to the process of a linear transformation in a vector space and a translation to another vector space in geometry. It keeps the two-dimensional graphics “flat”: straight after the transforming is still linear, “parallel”: the relative position relationship between the 2D graphics remain unchanged, parallel lines remain parallel lines, and the location of the order of points on a line [27].

An arbitrary affine transformation can be expressed as multiplied by a matrix, followed by a vector form.

Then, we can use affine transformation to represent the following three common forms of transformation:

(1)Rotation (linear transformation).

(2)Translation (vector plus).

(3)Scaling (linear transformation).

Affine transformation represents a mapping between two graphs, and we usually use the 2 x 3 matrix to represent affine transformations.

$$A=\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}_{2\times 2},\ B=\begin{bmatrix} b_{00} \\ b_{10} \end{bmatrix}_{2\times 1},\ M=[A \quad B]\begin{bmatrix} a_{00} & a_{01} & b_{00} \\ a_{10} & a_{11} & b_{10} \end{bmatrix}_{2\times 3} \tag{29}$$

Taking into account the use of matrix A and B for two-dimensional vector $X = \begin{bmatrix} x \\ y \end{bmatrix}$ transform, it can be expressed in the following form: $T = A \cdot \begin{bmatrix} x \\ y \end{bmatrix} + B$ or $T = M \cdot [x \quad y \quad 1]^T$

That is $T = \begin{bmatrix} a_{00}x + a_{01}y + b_{00} \\ a_{10}x + a_{11}y + b_{10} \end{bmatrix}$

Known M and X, want to get T. Apply the expression $T = M \cdot X$. The information about this association can be clearly expressed in matrix M because the matrix M is linked to two pictures, and the following is an example of the direct association of each of the three points in the two diagrams [28].

Fig. 16 embodies the schematic diagram of Affine transformation.

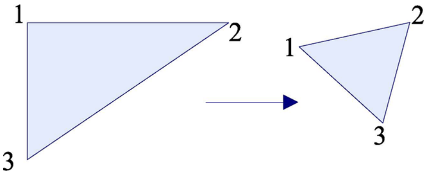


Fig. 16. Affine transformation schematic diagram

Among them, the points 1, 2 and 3 in left are mapped to the three points in right 2, and they still form triangles, but the shapes are not the same as before. We can obtain affine transformations by such two sets of three points, and then affine transformations can be applied to the actual scene Fig. 17.

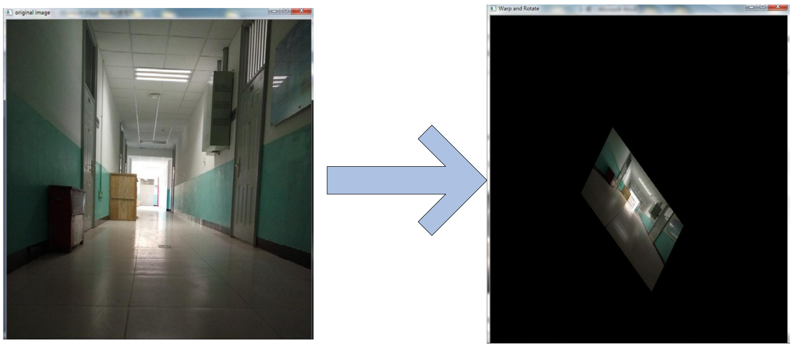


Fig. 17. Affine transformation

3.7 Hough Line Transform

Hough transform is a kind of image processing feature extraction technology, the process by calculating the cumulative result of local maximum value to a set according to the specific shape as Hof transform results in a parameter space [29].

Hough transform was first proposed by Paul Hough in 1962, Hough transform was originally designed to detect lines and curves, the first method is demanded to know the analytical equation of object boundary line, but does not require a priori knowledge of the relevant regional location. An outstanding advantage of this approach is the Robustness of the segmentation result, which is not very sensitive to incomplete or noisy data. After in 1972 by Richard Duda & Peter Hart to promote the use of the classical Hough transform is used to detect the line in the image, then Hough transform is extended to arbitrary object recognition, multi circle and ellipse. Using the Hough transform between the two coordinate space transformation will have the same shape in a space curve or straight line to form a mapping peak point of another space, so as to detect arbitrary shape problem into a statistical peak [30].

Hough line transformation is a method for finding the straight line. Before using Hough line transformation, the first step is to process the edge of image, that is to say, the direct input of Hough line transformation can only be the two valued image of the edge.

OpenCV supports three different lines of Hough transformation, they are: Standard Hough standard

Transform, SHT and Multi-Scale Hough Transform, MSHT, Progressive Probabilistic Hough Transform, PPHT.

Among them, the MSHT is a variant of the classical Hough transform (SHT) at multiple scales. The cumulative PPHT algorithm is an improved algorithm of Hough transform (SHT), which is the Hough transform in a certain range, separately calculate the line's direction and scope, so as to reduce the amount of calculation, shorten the calculation time, the PPHT is called "probability", because not all possible point accumulation accumulator in the plane, but the cumulative one part, the idea is that if the peak is high enough, only a small part of the time to find it, can substantially reduce the computation time [31].

The principle of Hough line transformation. A straight line can be represented by two variables in the two-dimensional space of the image:

(1) In Cartesian coordinates: can be represented by parameters: slope and intercept (m, b).

(2)In polar coordinates: can be represented by parameters: pole diameter and polar angle (r, θ).

For the Hough transform, we will use second polar coordinates to represent the straight line, the coordinates as the Fig. 18. Therefore, the expression of the straight line can be:

$$y = (-\frac{\cos\theta}{\sin\theta})x + (\frac{r}{\sin\theta}) \tag{30}$$

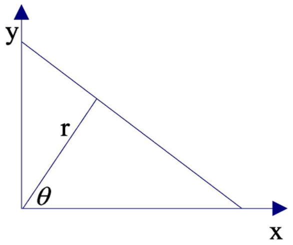


Fig. 18. Polar coordinates

Therefore, $r = x\cos\theta + y\sin\theta$

This means that each pair of (r_θ, θ) represents a straight line through the point (x_0, y_0).

Draw only the points $r > 0$ and $0 < \theta < 2\pi$ that satisfy the following criteria

The above indicates that a straight line can be detected by looking at the number of curves that are handed in at the plane $\theta - r$. And the more the curve at a point that means the line this intersection is represented by more points. In general we can set the threshold point on the line, how to define a curve at a point that we think the detection of a straight line.

This is the Hof transform line to do. It tracks the intersection between each point in the image. If the number of the corresponding curve at a point of the curve over the threshold, you can think this node represents the parameters of (r_θ, θ) in the original image as a straight line [32].

The principle of Hof gradient method.

(1)First edge detection of the image, such as using Canny edge detection.

(2)Then, for each non-zero point in the edge image, consider its local gradient, that is, the first derivative of Sobel in the X and Y directions is computed by the Sobel () Function, and the gradient is obtained.

(3)Using the gradient obtained, each point on the line specified by the slope is accumulated in the accumulator, where the slope is the distance from a specified minimum to the specified maximum.

(4)Simultaneously mark the location of each of the 0 pixels in the edge image.

(5)Then select the candidate centers from these points in the two-dimensional accumulator, each of which is larger than a given threshold and greater than all of its neighbors. These candidate centers are arranged in descending order so that the center of the most supporting pixel appears first.

(6)Next, for each center, consider all non 0 pixels.

(7)These pixels are sorted according to their distance from the center. From the maximum radius of minimum distance, from a choice of 0 pixels for the radius and most support.

(8)If a center receives an edge image that is not fully supported by 0 pixels and has sufficient distance to the selected center at first, it will be preserved.

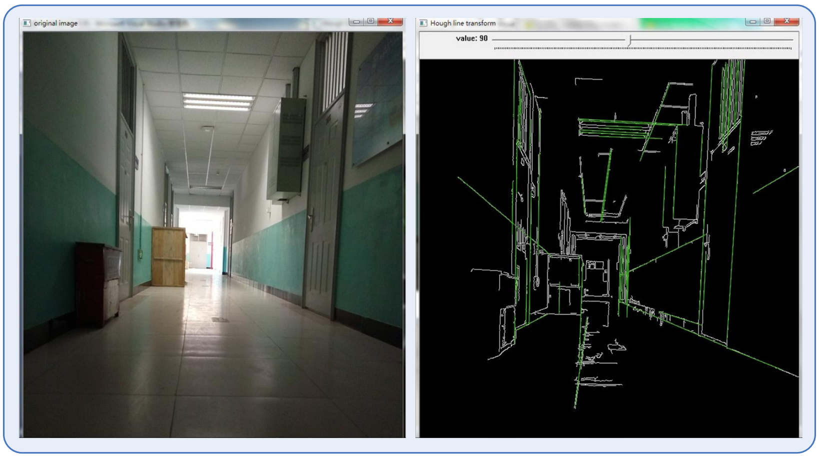


Fig. 19. Corridor scene Hough line transform

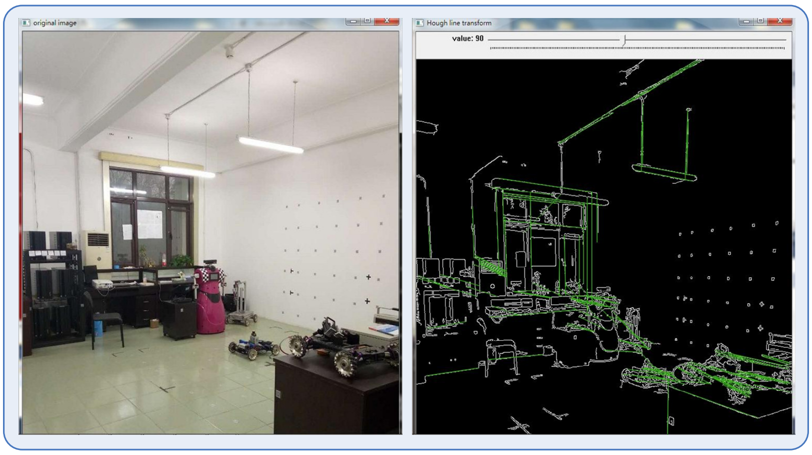


Fig. 20. Indoor scene Hough line transform

4 Experiment and Data Processing

In this paper, Microsoft Visual 2010 and OpenCV 2.4.9 are used as the developing platform to study the visual image processing of mobile robot. Image information is acquired by the vision sensor Kinect, the Canny edge detection, histogram equalization, Harris corner detection algorithm, sub-pixel corner detection, image affine transform, Hof transform processing line.

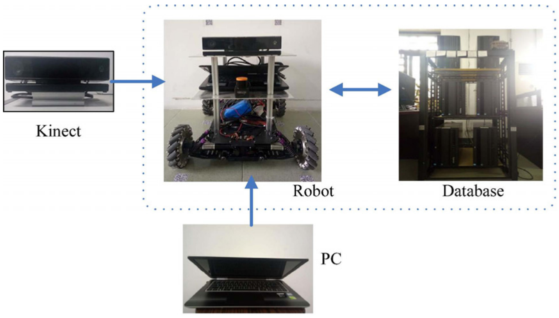


Fig. 21. Data flow diagram

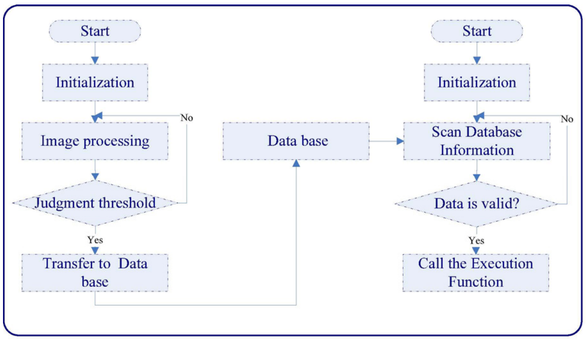


Fig. 22. Image processing schematic diagram

The processed image information can be better processed by the system. The data processing center in this paper adopts wireless transmission mode, establishes database in the server, and transfers the control information to the database through wireless module. The database adopts the SQL Server database, and the principle of applying SQL Server database for data transmission is to connect the upper computer, the server and the lower computer through the wireless module to form a lan. Then, tables are created in SQL Server to store the control information transmitted from the PC side. Send data in the end, the transmission control information to the scanning information to judge the decision after continuously to the server, the server processing will be transmitted to the robot operation instruction, complete the corresponding task.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61473027) and Beijing Key Laboratory for Biomimetic and Function of Robot Grant BZ0337.

References

[1] Y. Lin, H. Min, H. Zhou, F. Pei, A human-robot-environment interactive reasoning mechanism for object sorting robot, IEEE Transactions on Cognitive and Developmental Systems PP(99)(2017) 1-1

[2] F. Fabrizio, A. De Luca, Real-time computation of distance to dynamic obstacles with multiple depth sensors, IEEE robotics and automation letters 2(1)(2017) 56-63.

[3] F. Husain, H. Schulz, B. Dellen, C. Torras, S. Behnke, Combining semantic and geometric features for object class segmentation of indoor scenes, IEEE Robotics and Automation Letters 2(1)(2017) 49-55.

[4] R. Rahimi, C. Shao, M. Veeraraghavan, A. Fumagalli, J. Nicho, J. Meyer, S. Edwards, C. Flanningan, P. Evans, An industrial robotics application with cloud computing and high-speed networking, in: Proc. 2017 First IEEE International Conference on Robotic Computing (IRC), 2017.

[5] I. A. Seleem, S. F. M. Assal, Sliding mode control of underactuated five-link biped robot for climbing stairs based on real human data, in: Proc. 2017 IEEE International Conference on Industrial Technology (ICIT), 2017.

[6] M. Y. Cho, Y. S. Jeong, Human gesture recognition performance evaluation for service robots, in: Proc. 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017.

[7] Z. S. Hashemifar, K. W. Lee, N. Napp, K. Dantu, Consistent cuboid detection for semantic mapping, in: Proc. 2017 IEEE 11th International Conference on Semantic Computing (ICSC), 2017.

[8] Z. Ju, X. Ji, J. Li, H. Liu, An integrative framework of human hand gesture segmentation for human–robot interaction, IEEE Systems Journal PP(99)(2017) 1-11.

[9] L. Garrote, J. Rosa, J. Paulo, C. Premebida, P. Peixoto, U. J. Nunes, 3D point cloud downsampling for 2D indoor scene modelling in mobile robotics, in: Proc. 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2017.

[10] U. Asif, M. Bennamoun, F. A. Sohel, RGB-D object recognition and grasp detection using hierarchical cascaded forests, IEEE Transactions on Robotics 33(3)(2017) 547-564.

[11] L. Bolecek, P. Němec, J. Kufa, V. Ricny, Extending of the Kinect device functionality and the corresponding database, in: Proc. 2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA), 2017.

[12] H. C. Ravichandar, A. P. Dani, Human intention inference using expectation-maximization algorithm with online model learning, IEEE Transactions on Automation Science and Engineering 14(2)(2017) 855-868.

[13] A. Sento, P. Srisuk, Y. Kitjaidure, An intelligent system architecture for meal assistant robotic arm, in: Proc. 2017 9th International Conference on Knowledge and Smart Technology (KST), 2017.

[14] S. A. A. Shukor, M. A. A. Rahim, B. Ilias, Scene parameters analysis of skeleton-based human detection for a mobile robot using Kinect, in: Proc. 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2016.

[15] M. S. Nabipour, N. Arteghzadeh, S. A. A. Moosavian, A. Nasr, Visual servoing in a cable robot using Microsoft Kinect v2 sensor, 2016 4th International Conference on Robotics and Mechatronics (ICROM), 2016.

[16] W. Ye, Z. Li, C. Yang, J. Sun, C. Y. Su, R. Lu, Vision-based human tracking control of a wheeled inverted pendulum robot, IEEE Transactions on Cybernetics 46(11)(2016) 2423-2434.

[17] Y. L. Kuo, B. H. Liu, C. Y. Wu, Pose determination of a robot manipulator based on monocular vision, IEEE Access 4(2016) 8454-8464.

[18] S. B. Mane, S. Vhanale, Real time obstacle detection for mobile robot navigation using stereo vision, in: Proc. 2016 International Conference on Computing, Analytics and Security Trends (CAST), 2016.

[19] A. Aribowo, G. Gunawan, H. Tjahyadi, Adaptive edge detection and Histogram color segmentation for centralized vision of soccer robot, in: Proc. 2016 International Conference on Informatics and Computing (ICIC), 2016.

[20] F. Zhang, S. Zheng, Y. He, X. Shao, The research on attitude correction method of robot monocular vision positioning system, 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2016.

[21] C. Li, Q. Shi, C. Wang, Q. Huang, T. Fukuda, Calibration and implementation of a novel omnidirectional vision system for robot perception, 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2016.

[22] P. Andhare, S. Rawat, Pick and place industrial robot controller with computer vision, 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016.

[23] N. S. Simul, N. M. Ara, M. S. Islam, A support vector machine approach for real time vision based human robot interaction, in: Proc. 2016 19th International Conference on Computer and Information Technology (ICCIT), 2016.

[24] Y. Liu, T. Maruyama, A. Matsuzaka, Y. Matsumoto, A. Namiki, Assist control for ball catching using high-speed vision on a master-slave robot, in: Proc. 2016 IEEE/SICE International Symposium on System Integration (SII), 2016.

[25] K. Xia, Z. Weng, Workpieces sorting system based on industrial robot of machine vision, in: Proc. 2016 3rd International Conference on Systems and Informatics (ICSAI), 2016.

[26] G. Claudio, F. Spindler, F. Chaumette, Vision-based manipulation with the humanoid robot Romeo, in: Proc. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 2016.

[27] S. Huber, B. Selby, B. P. Tripp, Design of a saccading and accommodating robot vision system, in: Proc. 2016 13th

Conference on Computer and Robot Vision (CRV), 2016.

[28] L. Zhang, Z. Guo, H. Chen, L. Shuai, Corner-based 3D object pose estimation in robot vision, in: Proc. 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016.

[29] W. Shi, L. Zhang, Y. Yao, J. Zuo, X. Yao, Linear calibration for robot vision, in: Proc. 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016.

[30] X. Zhang, Y. Wang, Y. Fang, Vision-based moving target interception with a mobile robot based on motion prediction and online planning, in: Proc. 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2016.

[31] R. Tian, Q. Li, Research on the application of rectangle object constraint in active vision of mobile robot, in: Proc. 2016 International Conference on Robotics and Automation Engineering (ICRAE), 2016.

[32] Y. Zhu, X. Wang, B. Xu, Design of vision-based obstacle crossing of high-voltage line inspection robot, in: Proc. 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2016.