Research and Application of Image Edge Detection Algorithm Based on FOA

Xin XIE¹, Wei-ru WANG¹, Xun-yi JIANG¹, Tian-cheng WAN¹ and Feng-ping HU²

¹School of Information Engineering, East China Jiaotong University, Nanchang, China
²School of Civil Engineering, East China Jiaotong University, Nanchang, China

*Corresponding author

Keywords: FOA, Canny operator, Edge detection, Hilbert transform.

Abstract. Traditional differential-based edge detection algorithms have shortcomings such as inaccurate edge point positioning and corner point missing detection. This paper presents an application of FOA (Fruit Fly Optimization Algorithm) for establishing an image edge tracking model. Firstly, the prior knowledge of edge points is obtained by Canny edge detection operator. Then the corner points are extracted by Hilbert transform, and the edge tracking is accomplished by the coordination of the two mechanisms of randomness and positive feedback; Finally, a single pixel edge is obtained by using the forward movement mechanism and the loop termination condition. Compared with the original Canny edge detection algorithm, the precision and accuracy of image detection are better.

Introduction

Edge detection is the foundation of many other image processing techniques. Common edge detection operators include Roberts[1], Sobel[2], Prewitt[3], Laplacan[4], Kirsh[5], Hurckel[6], LOG[7], Canny[8], etc. Among them, Canny algorithm has better denoising ability and higher detection precision than other algorithms, and has a wide range of applications. However, the traditional Canny algorithm has problems such as parameters setting difficulty and poor detection effect in salt and pepper noise environment. In recent years, Canny algorithm has been improved by many scholars for these existing problems. Xuefang Chen[9] proposed an adaptive median filter to replace Gaussian filter, which has good adaptability to salt and pepper noise, however the operand is large, and the denoising effect of Gaussian noise is not good; Xinchao Xu[10] used a 3×3 convolution template to calculate the gradient, which increases noise immunity of the algorithm but it failed to make full use of the data of direction to calculate the gradient direction; Hongke Xu[11] introduced Otsu algorithm to calculate high and low thresholds according to the distribution of pixel gradient values, which increased the adaptability of the algorithm. However, traditional Otsu algorithm needs to traverse the gradient level of each pixel to determine the variance value between the largest categories, which has low efficiency. FOA (Fruit Fly Optimization Algorithm)[12], proposed by Wenchao Pan from Taiwan, China in 2012, is a swarm intelligence optimization algorithm based on the foraging behavior of fruit flies. The algorithm aims to improve the diversity, increase the searching space, and speed up the convergence rate of fruit fly individuals, by utilizing the keen olfactory and visual organs of fruit fly[13]. Qiwei Peng et al.[14] proposed an image segmentation method combining the improved two-dimensional Otsu method and fruit fly algorithm. By using the differential two-dimensional histogram, the joint probability is improved according to the size of gray level. Lixin Sun et al[15] proposed an image segmentation algorithm based on improved fuzzy mean clustering algorithm for fruit fly algorithm, which improves the accuracy of image classification and has strong robustness. However, it increases the computational complexity of the algorithm to a certain extent. Whether it meets the requirements of actual engineering for computational efficiency needs to be tested by practice.
In order to solve above problems, the Hilbert transform is used to extract undetected corner accordingly, then Canny algorithm can be employed to extract prior knowledge. Finally, we use FOA to locate the edges of the image. In this way, the established edge detection model has higher detection accuracy and faster algorithm implementation.

Traditional Canny Edge Detection Operator Analysis

1) In order to reduce the influence of noise, the image is smoothed by a two-dimensional Gaussian filter $G(x, y)$. Let the original image be $I(x, y)$, then the image after smoothing $H(x, y)$ can be expressed as:

$$H(x, y) = G(x, y) * I(x, y)$$

where: $G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$; “*” indicates convolution operation.

2) The gradient amplitude $A(x, y)$ and the direction $\theta(x, y)$ are calculated as follows:

$$
\begin{align*}
H_x(x, y) &= \frac{\partial G}{\partial x} * I(x, y) \\
H_y(x, y) &= \frac{\partial G}{\partial y} * I(x, y)
\end{align*}
$$

$$
\begin{align*}
A(x, y) &= \sqrt{[H_x(x, y)]^2 + [H_y(x, y)]^2} \\
\theta(x, y) &= \arctan \frac{H_y(x, y)}{H_x(x, y)}
\end{align*}
$$

3) Non-maximal suppression of gradient values. The specific method is: Mark the pixel point as a non-edge point by scanning an image, if gradient value of a pixel point is not greater than two adjacent pixel points in the gradient direction, otherwise, the point is marked as a candidate edge point.

4) Detect and join edges with a dual threshold algorithm. The specific method is: If the gradient amplitude of the pixel point is larger than the high threshold, it must be an edge point. If the gradient amplitude of the pixel is less than the low threshold, the point is definitely not an edge point. For a pixel that is between the high and low thresholds, investigate whether there is a pixel above the high threshold in the eight neighborhoods of the point. If it exists, the point is an edge point, otherwise it is a non-edge point.

In engineering practice, the image is always disturbed by random noise. so choosing the variance parameter $\sigma$ of the Gaussian function in the Canny edge detection algorithm is critical. The Gaussian filter is a low pass filter. Therefore, the larger $\sigma$ is, the narrower the pass band is, and the greater the inhibitory effect on higher frequency noise is, thus avoiding the detection of false edges, but it also smooths the edge of the signal, resulting in the loss of some edge points. On the contrary, the smaller $\sigma$ is, the wider the pass band is, the higher frequency details of the image can be detected, but the ability to suppress noise is relatively reduced, and false edges are prone to appear. An important goal of edge detection is to obtain the best compromise between edge location and noise filtering. The differential-based edge detection algorithm has inaccurate edge location and missed detection of corners. In order to solve the problem of corner miss detection. In this paper, the characteristics of signal Hilbert transform are analyzed, and the corner extraction algorithm based on Hilbert transform is proposed.

Traditional Canny Edge Detection Operator Analysis

Computers only can process and analyze discrete data. To analyze and process continuous signals, the
continuous signals are first sampled to obtain discrete signals. Let \( s[k] \) be a discrete signal whose Fourier transform is:

\[
S(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} s[k] e^{-j\Omega k}
\]  

(4)

To analyze the Hilbert transform of discrete signal \( s[k] \), \( s[k] \) can be used as an input to a discrete linear time-shift invariant system. The frequency response of the discrete linear time-shift invariant system is:

\[
H(e^{j\Omega}) = \begin{cases} 
-j & \text{for } 0 < \Omega < \lambda \\
0 & \text{for } \Omega = 0 \text{ and } \Omega = \lambda \\
j & \text{for } \lambda < \Omega < 2\lambda 
\end{cases}
\]  

(5)

After the discrete signal \( s[k] \) passes through the system, the frequency of the discrete output signal \( \hat{s}[k] \) is:

\[
\hat{S}(e^{j\Omega}) = H(e^{j\Omega})S(e^{j\Omega})
\]  

(6)

*Discrete Hilbert Transform (DHT)*[16] is an approximate ideal discrete Hilbert transform. The time domain of the Hilbert transform of the discrete signal \( s[k] \) can be expressed as \( \{s[k]\}, k = 0, 1, \ldots N-1 \), and:

\[
\{s[k]\} = IDFT\{H[n]S[n]\}
\]  

(7)

where \( \{S[n]\} \) is the *Discrete Fourier Transform (DFT)* of \( \{s[k]\} \), \( \{H[n]\} \) is a periodic discrete expression of \( H(e^{j\Omega}) \) in the formula (5).

\[
H[n] = -j\tilde{\text{sgn}}[n]
\]  

(8)

Let the discrete signal \( s[k] \) be \( N \) sampled data. If \( N \) is even, the periodicity of \( \tilde{\text{sgn}}[n] \) is a discrete sign function as:

\[
\tilde{\text{sgn}}[n] = \begin{cases} 
1 & \text{for } n = 1, 2, \ldots, N/2 - 1 \\
0 & \text{for } n = 0, N/2 \\
-1 & \text{for } n = N/2 + 1, \ldots, N-1 
\end{cases}
\]  

(9)

According to formula (6), \( \hat{s}[k] \) is obtained by circular convolution of \( s[k] \) and the following formula:

\[
h[k] = \frac{2}{N} \sum_{r=1}^{N/2-1} \sin \frac{2\pi rk}{N}
\]  

(10)

When \( N \) is an odd number, the periodicity of \( \tilde{\text{sgn}}[n] \) is a discrete sign function as:

\[
\tilde{\text{sgn}}[n] = \begin{cases} 
1 & \text{for } n = 1, 2, \ldots, (N-1)/2 \\
0 & \text{for } n = 0, N/2 \\
-1 & \text{for } n = (N+1)/2, \ldots, N-1 
\end{cases}
\]  

(11)
It is a two-dimensional image signal, and the Hilbert transform of the one-dimensional signal is generalized. Therefore, the two-dimensional discrete Hilbert transform of \( s[k_1,k_2] \) is transformed into a one-dimensional discrete Hilbert transform for implementation: Firstly, the columns of \( s[k_1,k_2] \) and \( h[k] \) are convoluted according to formula (7) and the Hilbert transform is gotten in the column direction. Then the convolution of the sum of the transform results to obtain the discrete Hilbert transform of the two-dimensional signal:

\[
\begin{align*}
\hat{s}[k_1,k_2] &= \sum_{r_1=0}^{N_1-1} \sum_{r_2=0}^{N_2-1} s[r_1,r_2] h_{2d}[k_1-r_1,k_2-r_2] \\
&= \sum_{r_1=0}^{N_1-1} \sum_{r_2=0}^{N_2-1} \{ \sum_{r_1=0}^{N_1-1} s[r_1,r_2] h[k_2-r_2] \} \times h[k_1-r_1]
\end{align*}
\]

The corner information of image signal in spatial domain is transformed into the extraction of extreme point by Hilbert transform; Neighborhood detection method is used to realize the detection of extreme points: Dividing the \( 3 \times 3 \) neighborhood of each pixel, and analyzing the magnitude relationship between the value of the intermediate pixel and the gray value of eight adjacent pixels in its neighborhood; If the value of the intermediate pixel is greater than the value of the surrounding 8 adjacent pixels, the intermediate pixel is considered to be a local corner, otherwise it is not a local corner.

**Image Edge Detection Based on Fruit Fly Optimization Algorithm**

When Fruit Fly are looking for food, individuals of the fruit fly firstly use their own olfactory organs to smell the food, and send odor information to surrounding fruit flies, or receive odor information from others. By comparing the location of the fruit fly that collected the best odor information in the current population, the fruit fly then uses its visual organs to inform others to fly to the location, and continue searching. Fig. 1 shows the foraging process of fruit flies.

In order to maximize the image edge detection efficiency, termination condition is needed for image edge search, even if the image edge search termination condition is not satisfied, the search is stopped and the detection result is outputted. In order to avoid image noise as the wrong edge point output, the data structure control idea uses the edge fuzzy mapping to complete the structure search result output. The movement rules of fruit flies on the image were set as random average movement to eliminate the influence of noise.

Data structure control is used to control the route selection of fruit flies, and any route will be selected in the image after extracting edge points and corner points. Firstly, each fruit fly is distributed on a \( M \times N \) two-dimensional grid lattice, where each grid point represents a 8-bit (1 byte) gray value.
of the $M \times N$ pixel (0-255). The corresponding pixels on the image, and the fruit fly will move on
this two-dimensional grid lattice, the location of the fruit fly at a certain moment and the direction of
movement of the fruit fly at this time represent the fruit fly itself. At each time, the location of the fruit
fly will have eight neighborhoods, which stipulate that the fruit fly can only be selected into one
adjacent square at each step, and cannot move more and more. The displacement matrix record is set
in the detection process to query the position, and the position is excluded in the next selection
process.

The definition function of the perceived intensity is $S_{(i,j)}$. The formula is as shown in (17). Its
function is to detect the structure of pixel points and judge whether the nearby pixel points are edge
pixels by similarity principle. The movement rules of fruit flies on the image are set to random
average movement, and a fuzzy variable $U$ is used to scan the structure search results. The fuzzy
variable scanned at time $t$ is $U_t$, and $\chi$ is the fuzzy control factor (usually set to $\chi = 4.2$ ). When the
structure search result satisfies the formula (13), the edge pixel points can be output.

$$U_{t+1} = \chi U_t (1 - U_t)$$

(13)

Let the line scan starting point $X_i^k$ of the fuzzy variable $U$ be:

$$X_i^k = p + (p - q) x_i^k$$

(14)

where: $x_i^k$ represents the blur vector of the fuzzy variable $U$ on the line $i$ of the pixel domain; $p$ is
the scan column; $q$ is the scan line.

Under the control of the fuzzy control factor $\chi$, $x_i^k$ satisfies:

$$x_i^{k+1} = \chi x_i^k (1 - x_i^k)$$

(15)

The termination condition of the image structure control idea for image edge search is
$\forall (m,n) \in x_i^k$. If the search result does not reach the termination condition, the fruit fly moving
iteration is continued with formula (16), where $X_j^k$ is the column scan starting point and $r,d$ is the
row and column moving condition respectively. The default fruit fly movement coordinate is $(-1,1)$, there are:

$$X_i^k = (X_i^k + r(-1,1), X_j^k + d(-1,1))$$

(16)

Taking the pixel domain as an example, the pixel point structure search route and the fruit fly
moving angle have $\theta$ off, as shown in Fig. 2, and 1, 2, 3, 4, and 5 in the figure indicate the setting of
the search route, according to the formula (17). The angle limit condition is used to determine the
perceived intensity of a certain coordinate point.

According to the setting of the search route in Fig. 2, the fruit fly will move along the edge of the
image, avoiding invalid iterations, and the perceived intensity $S_{(i,j)}$ can also be guaranteed to a higher
level. The perceived intensity $S_{(i,j)}$ and the search route for different fruit fly movement angles can be expressed by:
where $V$ represents the taste concentration of a certain coordinate point.

Canny operator can filter Gaussian noise effectively, which can locate edge accurately and completely compared to other operators, and the good connectivity is shown at the edge of image this detects. The Hilbert transform is used to extract corner information, and the corner and edge points are used as the heuristic information. The edge tracking is performed by the fruit fly optimization algorithm to obtain the single pixel edge. A specific flow chart of image edge detection based on FOA is shown in Fig. 3. FOA based image edge detection can be divided into the following steps.

**Figure 3. Edge detection flow chart.**

1. **Step1:** Use the principle of Canny edge detection operator to obtain the prior knowledge of edge points, and extract corner points by Hilbert transform;
2. **Step2:** Initialization algorithm parameters, set the displacement matrix in which move flies over the location of the record, in order to reduce the corresponding query time, initial taste concentration matrix;
3. **Step3:** The probability of fly direction was calculated and the current position was written into the displacement matrix.
4. **Step4:** Select the direction of the location of the fruit fly according to the optimal taste concentration, and the fruit fly group flies to the pixel to update the taste concentration information;
5. **Step5:** Repeat the above steps, terminate when the number of iterations reaches the maximum value or find the optimal solution, and mark the pixel;
6. **Step6:** Place the marker on white on the original image, and place the other points on black to smoothly connect the marker points to complete image edge detection.

Among them, in the initial position of fruit fly, the image is searched globally with the pixel points with higher probability as the starting point. The detection of high probability regions also applies the algorithm iterative process to the local edge search to, improve the efficiency of the algorithm; the addition of a taboo table helps strengthen the ability of fruit flies to find optimal solutions. The taboo table records the unknown information of the points that the fruit fly has selected in the table. When executing, the algorithm selects those points that have never been searched by selecting 8 adjacent positions for the pixel and skipping the points that exist in the taboo table, so that a larger area of search target can be achieved and the meaningless round-trip movement of the fruit fly within the local search range is avoided. The edge tracking model of fruit fly algorithm achieves the guiding...
effect of taste concentration and heuristic information on edge tracking, avoiding the distribution and walking of fruit flies in non-edge areas.

**Experimental Simulation and Results Analysis**

In order to test the feasibility of the algorithm, experiments were conducted. The experimental environment is *Intel Core (TM) I5 2.50GHz with 4GB RAM*; the simulation software is *Matlab R2016a*.

Experiment 1. Comparison of edge detection effect of standard circle

Firstly, Matlab software can be used to draw a standard unit circle. The resolution of the picture is 500×500 pixels, and the diameter of the circle is 377.0 pixels. Four endpoints can be taken on the circle as the leak detection points, which are A, B, C and D respectively. Sub-pixel coordinate detection was performed for the four points. The detection results are shown in the figure below, which can output the circular edges carefully and continuously. The test results are shown in Table 1.

![Standard round](image1)

![Standard circle edge detection](image2)

Table 1. Standard round edge detection results.

|                | A                  | B                  | C                  | D                  |
|----------------|--------------------|--------------------|--------------------|--------------------|
| Pixel level coordinates | (254.0,444.0)     | (254.0,67.00)     | (65.50,255.50)    | (442.50,255.50)    |
| Sub-pixel coordinate   | (253.20,442.50)   | (253.20,66.20)    | (65.00,254.35)    | (441.00,254.35)    |
| Coordinate offset       | (0.80,1.50)       | (0.80,0.80)       | (0.50,1.15)       | (1.50,1.15)       |
| Measuring diameter length | 376.30 pixel     | 376.00 pixel      |                   |                   |
| Absolute error         | 0.7 pixel          | 1.0 pixel          |                   |                   |
| Relative error         | 0.186%             | 0.266%             |                   |                   |

As shown in Table 1, the accuracy of the algorithm can reach the level of 1 pixel, with relatively small error and high measurement accuracy. The above experiments verify the correctness of the algorithm in this paper.

Experiment 2 compares it with standard Canny operator and *ACO* algorithm

Canny edge detection algorithm is a widely used edge detection operator. The algorithm in this paper, the standard Canny algorithm and the traditional image edge detection algorithm based on *Ant Colony Optimization (ACO)* were respectively applied to the three images of Lena (512×512), Peppers (256×256) and Baboo (512×512), as shown in Fig. 6: For noiseless images, Canny's ability to retain detailed edges and high-frequency components is weaker than the algorithm in this paper. The traditional image edge detection algorithm based on ant colony algorithm is not far behind the algorithm in this paper. Each edge point in the detected edge image is scanned, The point is marked as...
the exact edge point if it is the edge point of the corresponding position in the image detected by the Canny algorithm. Then, the number of missed edges detection is the difference between the total edge point of the detected image and the accurate edge point, the number of false detection edges is the difference between the edge point of the detected image and the edge point of the standard image. The false detection rate is the ratio of the number of false detections to the total pixels of the image; the missed detection rate is the ratio of the number of missed detections to the total pixel of the image; and the positioning error is the sum of the missed detection rate and the false detection rate. The test results are shown in Table 2:

![Image of comparison](image.png)

Figure 6. Comparison of noise-free edge detection algorithms.

| Image  | Algorithm | Rule number | The rule rate (%) |
|--------|-----------|-------------|------------------|
| Lena   | FOA       | 14221       | 0.2171           |
|        | ACO       | 15743       | 0.2244           |
|        | Canny     | 25875       | 0.3683           |
|        | FOA       | 1876        | 0.0357           |
| Peppers| ACO       | 2056        | 0.0413           |
|        | Canny     | 2984        | 0.0617           |
|        | FOA       | 14553       | 0.1041           |
| Baboon | ACO       | 17995       | 0.1373           |
|        | Canny     | 22549       | 0.1723           |

It can be seen from Table 2 that the missing rate of image edge detection under different algorithms is different. Because the edge obtained by Canny algorithm is used as a standard edge point in this paper, and the edge obtained by Canny algorithm is usually more complete and may appear false edge, the edges detected by the algorithm proposed in this paper are all real edges, but the edges are obviously incomplete. Although the missing rate of the algorithm is relatively high at this time, the error rate is 0, the overall positioning error of the algorithm in this paper is only reflected in the miss
rate, and the positioning error in this paper is the miss rate. In the table, the value of the algorithm in this paper is the smallest, which indicates that the edge detection and extraction effect of the algorithm in this paper is better.

Dong Hu[17] introduced a method for evaluating the degree of edge connectivity: If \( NP \) indicates the number of detected edge points and \( NL \) indicates the number of 4 links in the edge image, the \( NL/NP \) value indicates the degree of edge linear connection. The effect of the degree of linear connection of the edge on the overall edge evaluation is reflected in the false detection and missed detection of the edge. When the number of false detections and missed detections of the edge points are large, the edges with poor connectivity are displayed, vice versa. Therefore, it is reflected in the table that the smaller the value \( NL/NP \) is, the better the edge linear connection is, the better the edge effect is extracted. Comparison of algorithm performance is shown in Table 3 below.

| Image | NP  | NL  | NL/NP | NP  | NL  | NL/NP | NP  | NL  | NL/NP |
|-------|-----|-----|-------|-----|-----|-------|-----|-----|-------|
| Lena  | 2057| 83  | 0.0404| 2014| 67  | 0.0332| 1975| 46  | 0.0233|
| Peppers | 14738| 77  | 0.0052| 9763| 43  | 0.0044| 7624| 9   | 0.0012|
| Baboon | 1512| 384 | 0.2540| 1506| 69  | 0.0458| 10572| 132 | 0.0125|

From the ratio of the linear connectivity degree to the quadratic number (\( NL/NP \)), the smaller the ratio is, the better the linear connection will be. It can be seen from Table 3 that the \( NL/NP \) of the edge detection map obtained by the fruit fly optimization algorithm is much smaller than the value of the traditional Canny algorithm, which indicates that the edge linear connection obtained by the fruit fly optimization algorithm is effective.

Table 4 records the time taken by the above three groups of pictures and three algorithms. It can be seen from Table 4 that the time taken by the algorithm in this paper is longer than that taken by Canny operator method, but it has certain advantages in time complexity compared with ACO algorithm.

| Image     | FOA(s) | ACO(s) | Canny(s) |
|-----------|--------|--------|----------|
| Lena (512x512) | 0.674  | 0.973  | 0.320    |
| Pepper (256x256) | 0.620  | 0.935  | 0.305    |
| Baboon (512x512) | 0.701  | 0.966  | 0.318    |

Conclusion

In this paper, Canny edge detection operator is used to get the prior knowledge of edge points by using the different proportion of pixel gray and corner points in the image as the inspiration information. Then an edge tracking model based on \( FOA \) is established. The single-pixel edge of the image can be obtained by using random average movement mechanism and cyclic termination condition, which can avoid the distribution and search of drosophila in the non-edge region. The experimental results show that compared with Canny edge algorithm, the method in this paper has better performance in image edge detection and higher precision in edge detection. However, the algorithm is slower than the traditional differential-based edge detection algorithm. The step size setting of fruit fly needs further improvement to reduce the search time of edge information.

Acknowledgements

This work is supported by the National Natural Science Foundation of China, under Grant Nos. 61762037, 61872141, and Natural Science Foundation of Jiangxi Province, under Grant Nos 20181BAB206037.
References

[1] Yuan W Q, Li D S. Edge detection based on directional space[J]. Frontiers of Electrical & Electronic Engineering in China, 2006, 1(2):135-140.

[2] Casarrubias-Vargas H, Pettrilli-Barceló A, Bayro-Corrochano E. Fast Edge Detection in RGB-D Images[C]// Iberoamerican Congress on Pattern Recognition. 2014:868-875.

[3] Lipkin B S, Rosenfeld A. Picture Processing and Psychopictorics[J]. Transactions of the American Microscopical Society, 1970, 91(2):99-105.

[4] Haralick R M. Digital Step Edges from Zero Crossing of Second Directional Derivatives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, PAMI-6(1):58-68.

[5] Wen L U, Zhang G Y, Cheng Y F, et al. An Improved Kirsch Edge Detection Method[J]. Information of Medical Equipment, 2007, 22(12):10-12.

[6] Hueckel M H. A Local Visual Operator Which Recognizes Edges and Lines[J]. Journal of the Acm, 1973, 20(4):634-647.

[7] Hui K, Sarma S E, Feng T. Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(1):408-418.

[8] Canny J. A computational approach to edge detection [M]//Readings in Computer Vision. 1987: 184-203.

[9] Chen, Xue Fang, Guan, Hao Bing, Gu, Ji Nan, et al. A Study and Improvements on Canny Algorithm[J]. Advanced Engineering Forum, 6-7:205-209.

[10] Xu Xinchao, Fu Chen, Xu Aijong. An Improved Canny Edge Extraction Method[J]. Remote Sensing Information, 2016, 31(5):43-46.

[11] Xu Hongke, Qin Yanyan, Chen Huiru. An edge detection algorithm based on improved Canny[J]. Infrared Technology, 2014, 36(3):210-214.

[12] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26: 69-74.

[13] Hu Nengfa. Evolutionary Drosophila Algorithm and Its Application[J]. Computer Technology and Development, 2013, 23(7):131-137.

[14] Peng Qi-wei, Luo Wang, Feng Min, et al. Image segmentation method based on improved two-dimensional Otsu method and Drosophila algorithm[J]. Journal of Computer Applications, 2017(A02):193-197.

[15] Sun Lixin, Zhang Yizhi, Deng Xianrui, et al. Image Segmentation Based on Adaptive Drosophila Algorithm Optimized Fuzzy Mean Clustering Algorithm[J]. Control Engineering, 2016, 23(4): 494-499.

[16] Noo, Frédéric, Clackdoyle R, Pack J D. A two-step Hilbert transform method for 2D image reconstruction[J]. Physics in Medicine and Biology, 2004, 49(17):3903-3923.

[17] Hu D, Tian X. A Multi-Directions Algorithm for Edge Detection Based on Fuzzy Mathematical Morphology[C]//International Conference on Artificial Reality & Telexistence-workshops. IEEE, 2006:361-364.