ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

Picture Book Page-Turning Detection Algorithm Based on Perceptual Hash

Gao Qun

School of Intelligent Transportation Modern Industry, Anhui Sanlian University, Hefei, 230601, China

Abstract: *Through the analysis of the requirements for page-turning detection algorithms on smart reading devices, a picture book page-turning detection algorithm based on perceptual hashing is designed and efficiently implemented in C language. The principle of the perceptual hashing algorithm is detailed, along with the implementation process and threshold update strategy. Experimental analysis is conducted to evaluate the effectiveness of the algorithm, and key parameters are selected and optimized. The experimental results show that the algorithm performs well both on general-purpose PCs and embedded devices. On a manually annotated test set, the detection rate reaches 94.5%, providing an implementation idea for page-turning detection on devices. Future work will focus on optimizing performance in special scenarios and improving the robustness of the algorithm.*

Keywords: *Picture book page-turning detection; Perceptual hashing; Detection rate; Smart reading devices*

1. Introduction

In today's digital and intelligent era, the popularity of intelligent reading devices such as e-readers and tablet PCs has brought people a brand new reading experience. These devices not only have the advantage of convenient storage and carrying, but also can enhance the fun and interactivity of reading through rich multimedia elements. Especially for children's books, such as picture books, the application of smart reading devices provides a more vivid and attractive way for children to read.

Intelligent reading devices generally use the built-in camera to take continuous pictures, automatically detect the user's actions such as changing books and turning pages, and automatically obtain the photos that need to be recognized, so the page turn detection of picture books has become an important research topic. The accuracy and timeliness of picture book page turn detection directly affects the smoothness and experience of children's reading. Effective page turn detection of picture books can realize automatic page switching, playing related audio or animation, etc., creating a more immersive reading environment for children.

In practice, picture book page turn detection faces many challenges. For example, different reading scenes (e.g., changes in indoor light, differences in reading angle) may lead to changes in the quality and characteristics of image acquisition; children may turn the page quickly or incompletely when reading, which requires the detection algorithm to have high sensitivity and accuracy; in addition, in order to ensure real-time, the detection algorithm needs to complete the processing in a shorter period of time in order to avoid lags or delays.

The perceptual hash algorithm, as an efficient method for image feature extraction and similarity comparison, has achieved remarkable results in the fields of image recognition and content retrieval [1-7]. Applying the perceptual hash algorithm to page turn detection in picture books is expected to solve the above faced challenges and improve the accuracy and efficiency of detection.

In this paper, we propose a page turn detection algorithm for picture books based on perceptual hashing, aiming to provide more accurate and reliable page turn detection service for picture books reading in smart reading devices, and to enhance children's reading experience.

2. Technical Principle

In intelligent picture book reading devices, there are currently two ways for the client to obtain photos through the built-in camera:

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

1) Continuous camera, automatically detecting the user's operations such as changing books and turning pages, and automatically obtaining the photos that need to be retrieved;

2) Timed camera, which gets multiple pictures per unit of time, retrieves them all, and gets the best candidate from them.

The following algorithm mainly focuses on the first method, and proposes a feasible page flip detection i.e. video motion detection scheme.

Perceptual Hash Algorithm (PHA) is a collective term for a class of hash algorithms, which are mainly used in the search work of similar images. It can convert the features of a picture into a set of “fingerprint” strings (hash values), by comparing these fingerprints to determine the similarity of the picture.

2.1 Mean hash

Mean hash algorithm mainly uses the low frequency information of the picture, and its process mainly contains reducing the size, simplifying the color, calculating the average value, comparing the gray value of the pixels and calculating the hash value.

The hash value is not much affected for pictures that are enlarged or reduced, change the aspect ratio, increase or decrease the brightness or contrast, change the color, and so on.

The similarity of two images can then be measured by the Hamming distance of their hash values: first calculate the hash fingerprints of these two images, which is the 64-bit 0 or 1 value, and then calculate the number of different bits (Hamming distance). If this value is 0, it means that these two images are very similar, if the Hamming distance is less than 5, it means that they are somewhat different but more similar, if the Hamming distance is greater than 10, it means that they are completely different images.

2.2 Enhanced algorithm: pHash

The mean hash, although simple, is very much affected by the mean value. For example, gamma correction or histogram equalization of an image affects the mean value and hence the final hash value. There exists a more robust algorithm called pHash, which takes the mean value approach to the extreme. Use Discrete Cosine Transform (DCT) to get the low frequency components of the image.

Discrete Cosine Transform (DCT) is an image compression algorithm that transforms an image from pixel domain to frequency domain. Generally images have a lot of redundancy and correlation, so after transforming to the frequency domain, only a very small portion of the coefficients of the frequency components are not zero, and most of the coefficients are zero (or close to zero).

The flow of the pHash algorithm [8-9] is as follows:

1) Reduce the size: pHash starts with small images, but images larger than 8*8, 32*32 is the best. The purpose of this is to simplify the DCT calculation, not to reduce the frequency.

2) Simplify color: Convert the picture into a grayscale image to further simplify the calculation.

3) Calculate DCT: Calculate the DCT transform of the picture to get a 32*32 matrix of DCT coefficients.

4) Shrink DCT: Although the result of DCT is a 32*32 size matrix, we just need to keep the 8*8 matrix in the upper left corner, which presents the lowest frequency in the picture.

5) Calculate the mean value: just like the mean hash, the mean value of the DCT is calculated.

6) Calculate the hash value: this is the main step, according to the 8 * 8 DCT matrix, set 0 or 1 of the 64-bit hash value, greater than or equal to the average value of the DCT is set to “1”, less than the average value of the DCT is set to “0”. Combined together, it constitutes a 64-bit integer, which is the fingerprint of this picture.

The result does not tell us the low frequency of authenticity, but only a rough idea of the relative proportion of the frequency relative to the mean. The advantage of the perceptual hash algorithm is that the operation speed is extremely fast, and no matter how the picture's height and width, brightness or even color changes, as long as its overall structure remains unchanged, the generated hash value will not change, which makes it in the field of image recognition, similar picture search and other fields have a wide range of applications.

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

For example, in the image search engine, the perceptual hash algorithm can be used to quickly screen out images similar to the target image; in image copyright protection, the hash value of the image can be calculated to establish an image fingerprint library, which can be used to detect the existence of copyright infringement and so on [6]. At the same time, the algorithm also has certain limitations, for example, for some images that have undergone complex transformations or have small content differences, it may be difficult to accurately distinguish their similarity. In practical applications, it is usually necessary to combine with other algorithms or methods to improve the accuracy and effect.

3. Algorithm realization

The algorithm mainly obtains the characteristic parameters of the image to be detected, and determines the state of the image to be detected according to the characteristic parameters, where the state refers to the page-turning state and stable state, the page-turning state indicates that the image to be detected is still in the user's page-turning action, and the stable state identifies that the image to be detected is a stable image after the user performs page-turning, and the stable state image can be used for the subsequent further operations of the intelligent reading device, such as image recognition [7-9].

3.1 Algorithmic flow

The overall framework flowchart of the algorithm is shown in Figure 1.

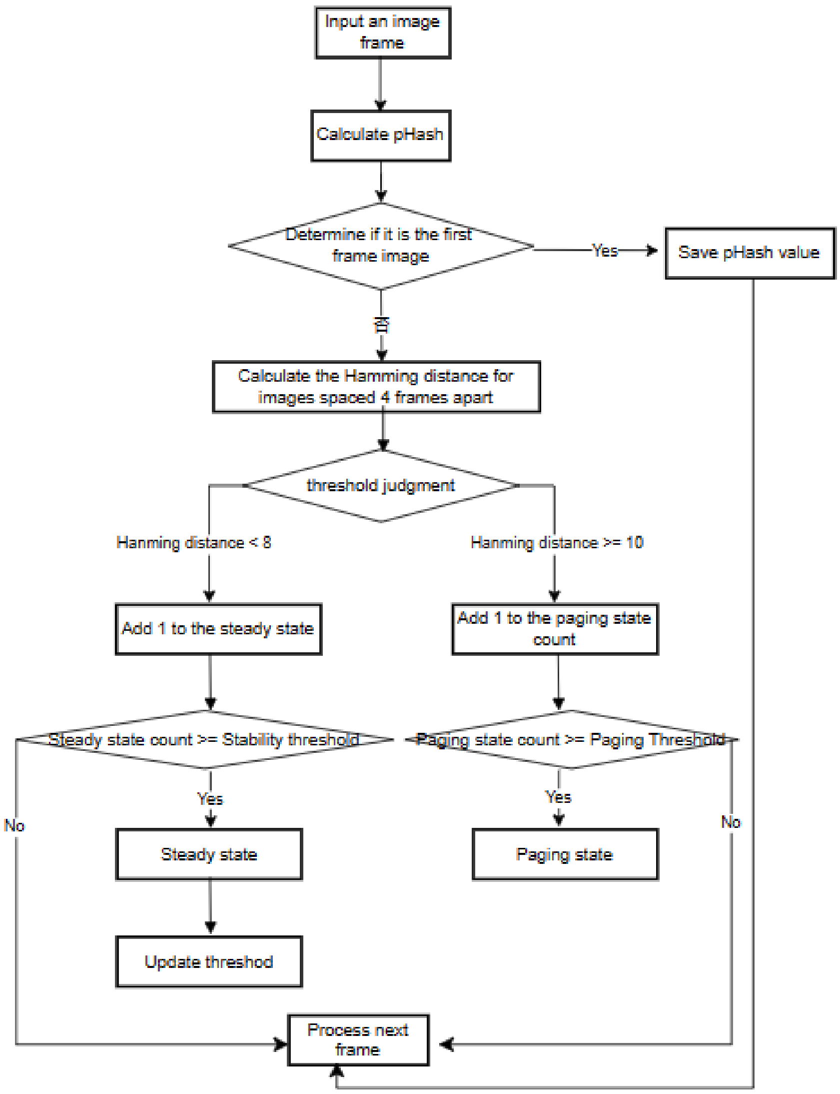

Figure 1: Flowchart of page-flip detection algorithm

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

The core code of the algorithm is shown in Figure 2.

```
//Page turning detection and judgment
if ((fMeanValue <= pEngine->thr_page) && (fMaxValue <= pEngine->thr_page + 2)){
    //Calculate the threshold after page turning is stable.
    if (pEngine->nPageFrame >= THR_PAGE_NUM){
        if (pEngine->nStableFrame < STABLE_PAGE_NUM){
            tcInt32 nFrameIdx = pEngine->nStableFrame % STABLE_PAGE_NUM;
            pEngine->min_mvalue[nFrameIdx] = fMinValue;
            pEngine->mean_mvalue[nFrameIdx] = fMeanValue;
            pEngine->max_mvalue[nFrameIdx] = fMaxValue;}
        else if (pEngine->nStableFrame == STABLE_PAGE_NUM){
            //Calculate the threshold after each page turning is stable.
            tcFloat32 fThrMinSum = 0, fThrAveSum = 0, fThrMaxSum = 0;
            pEngine->min_mvalue[STABLE_PAGE_NUM] = fMinValue;
            pEngine->mean_mvalue[STABLE_PAGE_NUM] = fMeanValue;
            pEngine->max_mvalue[STABLE_PAGE_NUM] = fMaxValue;
            for (tcInt32 nIdx = 0; nIdx < STABLE_PAGE_NUM + 1; nIdx++){
                fThrMinSum += pEngine->min_mvalue[nIdx];
                fThrAveSum += pEngine->mean_mvalue[nIdx];
                fThrMaxSum += pEngine->max_mvalue[nIdx];}
            //Update the threshold.
            pEngine->thr_page = ((2 * fThrMaxSum) - fThrAveSum)/(tcFloat32)(STABLE_PAGE_NUM + 1);
            pEngine->threshold_stable = pEngine->thr_page-fThrMinSum/(tcFloat32)(STABLE_PAGE_NUM + 1);
            pEngine->nPageFrame = 0;
            pEngine->nStableFrame++;
            pEngine->m_nBookStatus = BOOK_STATUS_STABLE;
            return BOOK_STATUS_STABLE;}}
    else{//initial stable state, there is no need to update the threshold.
        if (pEngine->nStableFrame == STABLE_PAGE_NUM){
            pEngine->nPageFrame = 0;
            pEngine->nStableFrame++;
            pEngine->m_nBookStatus = BOOK_STATUS_STABLE;
            return BOOK_STATUS_STABLE;}}
    pEngine->nStableFrame++;
}
else if (fMeanValue > pEngine->thr_page){
    pEngine->nStableFrame = 0;
    if (pEngine->nPageFrame >= THR_PAGE_NUM){
        pEngine->m_nBookStatus = PAGING_STATUS;}
    pEngine->nPageFrame++;
}
```

Figure 2: Core code of page-flip detection algorithm

3.2 Updating of Threshold Threshold Values

As shown in the above algorithm flow, after each detection of the stable state, the update of the page-flip state and the stable state threshold value is performed, and the updated value is used as the subsequent detection.

The following influencing factors are mainly considered for updating the page flip threshold:

1) Imaging parameters. This mainly includes parameters such as image tilt angle and brightness of the imaging environment;

2) Parameters that can characterize the size of the page flip possibility for N consecutive images;

3) Parameters that characterize the speed of page turning.

The following influencing factors are mainly considered for updating the stabilization threshold:

1) The page turn threshold value determined above;

2) The parameter that characterizes the probability size of the steady state for N consecutive images;

3) Parameters that characterize the speed of page turning.

The key parameters such as the mean value of the average value of the Hamming distances of the current image to be detected and the M consecutive frames of the image in front of it, the mean value of the maximum Hamming distance, and the mean value of the minimum Hamming distance are mainly taken into account in the calculation of the Hamming distance.

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

4. Experimental results and analysis

4.1 Evaluation indexes and experimental data

The evaluation of the effect for the picture book page turn detection algorithm is mainly evaluated from the following indicators:

(1) Detection rate: refers to the number of times the algorithm detects one clear image after the user has performed a page-turning action, that is, the stable state image mentioned by the above algorithm. The detection rate is equal to the number of steady state images detected by the algorithm divided by the total number of all steady state images. (Explain that for every page turn that occurs after the user may produce a number of consecutive clear images, these images are counted as a stable image, the detection results belong to any one of them are considered to be detected correctly).

(2) multi-detection rate: such as the above users may produce a number of consecutive clear images for each page action, for these images if the algorithm gives a number of stable state, it is considered to be more detected, which will increase the number of subsequent image recognition and detection of intelligent reading devices, will increase the power consumption of the device and may lead to the risk of intelligent reading devices as a whole can not be in real time, so the project is also an important evaluation index of the page detection algorithms. So this item is also an important evaluation index of the page-flip detection algorithm.

Experiments using the image shown in Figure 3.

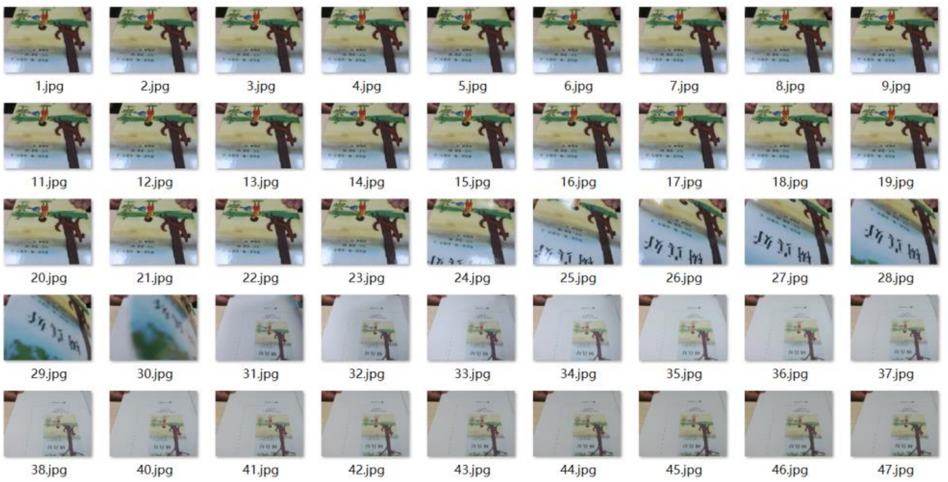

Figure 3: Example of page-turning detection map

The test image is a 12-megapixel camera with a maximum resolution of 1920*1080, simulating the camera position placement of the intelligent reading device, 10 people were selected to complete the video capture of the page-turning action of 88 picture books, and the captured video was later processed in accordance with the 30fps for the video-to-image sequence. The experimental data are shown in Table 1.

Table 1: Test data set

Data set	Total number of images	Number of stabilized images to be detected
Commonly Illustrated Books	82896	1890

4.2 Threshold threshold test

For normal page turning and fast page turning, the Hamming distance between the two frames before and after is greater than 10 and lasts for more than 3 frames when in the page turning state; while in the slow motion page turning, it is difficult to determine the page turning state by the Hamming distance between the two frames before and after, because the image changes are not obvious enough. Therefore, we consider comparing the Hamming distance of image hash fingerprints at intervals.

Calculate the Hamming distance of 4 frames apart, and get the results as in Figure 4:

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

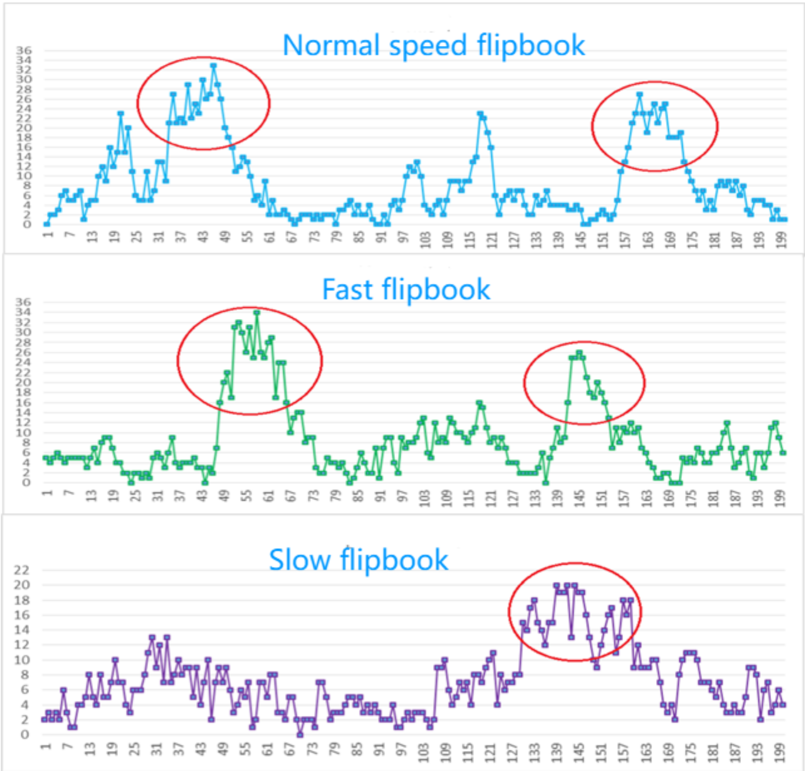

Figure 4: Hamming distance for images differing by 4 frames

At this time, the page-turning action in all three cases can be clearly observed, and it is known through experiments that the Hamming distance of 4 frames difference is more accurate for judging the page-turning action.

The following values were obtained from the experiments to determine the conditions of the page-turning state and the stable state:

1) Page turning: the Hamming distance between the two frames before and after is >=10 for 10 frames;

2) Stable: the Hamming distance between the front and back images is <8 for 5 frames.

4.3 Algorithm Effect

By implementing the page-flip detection threshold adjustment strategy in multiple groups, the improvement of the recognition effect is shown in Table 2.

Table 2: Algorithm effect test

	Baseline		Optimized version	
Test set	Detection Rate	Multiple Detection Rate	Detection Rate	Multiple Detection Rate
88 books	66.29%	3.44%	94.54%	1.31%

4.4 Algorithm Performance

In order to compare the performance of the page-flip detection algorithm, two platforms were selected for testing and the results are shown in Table 3.

Table 3: Algorithm efficiency test results

Target Platform	Processor	Memory required for algorithms	Average time spent per image detection (in milliseconds)
PC	12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz	374KB	1.1
MT8167A	Cortex-A35 1.5G quad-core	374KB	2.4

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

4.5 Analysis of results

This experiment provides a detailed evaluation of the efficiency and accuracy of page flip detection. By testing the performance of the algorithm under different platforms and verifying the accuracy on multiple datasets, the following analysis results can be obtained:

1) Algorithm efficiency related analysis

The tests of the algorithm on two different hardware platforms show that on a general-purpose PC, the average time consumed for each image detection is only 1.1 ms, which is attributed to its powerful computational capability and high processor frequency, while the detection time consumed by MediaTek MT8167A platform, despite its not-so-low configuration, increases significantly to 2.4 ms. This shows that the performance of the algorithm on a chip with less processing power needs to be further optimized. However, the overall performance performance are better; in addition, the algorithm memory occupation is independent of the platform, only 374KB is needed, showing good memory management.

2) Algorithm Accuracy Analysis

In the accuracy test, the results of stabilized images after manually labeled page turns were used as a benchmark to ensure the accuracy and authority of the evaluation. The test results show that the detection rate can reach 94.5%, the multi-detection rate is controlled within 2%, and the overall algorithm effect reaches usability.

The reasons for the poor recognition effect on individual test sets are analyzed as two main reasons: one is that the individual picture book images are too single, resulting in poor feature extraction; the other is that the recognition effect and the camera position, the angle of the user's book placement, the speed of page turning, and the ambient brightness at the time of the image acquisition have a greater relationship, and need to be further optimized.

Comprehensively analyzing the above, the drawing page turn detection algorithm in this study shows efficient and relatively accurate recognition ability. However, there is still room for improvement in the effect for special scenes. Future work can focus on improving the robustness of the algorithm and further enhancing the effect.

5. Conclusion

Through in-depth research and practice on the page turn detection algorithm for picture books, an efficient and accurate detection method has been proposed, and the method has been designed and implemented in detail. The test on the two platforms shows that the algorithm has a fast processing speed while ensuring a high accuracy rate, which can meet the real-time requirements of intelligent reading devices.

The experiments also show that there is still room for improvement in the current work. In particular, the improvement of the effect when the camera captures images with poor position and the improvement of robustness of different platforms will be the focus of future work in the expectation that the algorithm can be applied more effectively on smart reading devices.

Acknowledgements

Funded projects: Anhui Sanlian University, 2024 School-level Research Platform Key Project: Application and Optimization of Chinese-English Segmentation Algorithm on Intelligent Devices (Project No. PTZD2024013)

References

[1] Cao F, Yao S, Zhou Y, et al. Perceptual authentication hashing for digital images based on multi-domain feature fusion [J]. Signal Processing, 2024, 223, 109576-109576.
[2] Xu Mengqin, Zhou Haibo, Yin Jinming. Self-location and fault judgment of mobile robot based on Perceptual hash[J/OL].Journal of Tianjin University of Technology, 1-5[2024-07-28]. http://kns.cnki.net/kcms/detail/12.1374.n.20240325.1908.004.html.
[3] Zhou Yuanding, Fang Yaodong, Qin Chuan. Large-scale image dataset for perceptual hashing[J]. Journal of Image and Graphics, 2024, 29(02):343-354.

ISSN 2616-5775 Vol. 7, Issue 10: 8-15, DOI: 10.25236/AJCIS.2024.071002

[4] Qin C, Hu Y, Yao H, et al. Perceptual Image Hashing Based on Weber Local Binary Pattern and Color Angle Representation. [J]. IEEE Access, 2019, 7 45460-45471.
[5] Ma Bin, Wang Yili, Xu Jian.A Bidirectional Generative Adversarial Network-Based Perceptual Hash Algorithm for Image Content Forensics [J]. Chinese Journal of Computers, 2023, 46(12):2551-2572.
[6] Moumita R, Meitei D T, Shyamosree P. A perceptual hash based blind-watermarking scheme for image authentication [J]. Expert Systems with Applications, 2023, 227
[7] Li Wenju, Wang Zijie, Cui Liu. Improved Cam Shift tracking algorithm based on SIFT and perceptual hash [J].Electronic Measurement Technology, 2023, 46(04):184-192. DOI:10.19651/j.cnki.emt.2210524.
[8] Huo Delu, Yuan Chao, Li Jialong. Fingerprint Image Perceptual Hashing Algorithm Based on Relative Position of Feature Points[J].Industrial Control Computer, 2022, 35(03):52-53+85.
[9] Guo Shuangle, Zhang Jianguang. Analysis of Video Motion Recognition Technology Based on Accelerated Robust Features [J].Electronic Technology, 2023, 52(11):158-159.