

Retraction

Retracted: Application of Embedded Intelligent Target Detection System in Basketball Movement Test

Wireless Communications and Mobile Computing

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external

researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] C. Zhang and Y. Wu, "Application of Embedded Intelligent Target Detection System in Basketball Movement Test," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 5484706, 14 pages, 2022.

Research Article

Application of Embedded Intelligent Target Detection System in Basketball Movement Test

Chi Zhang^{1,2} and Yuanyuan Wu³ 

¹Ministry of Public Sports, Modern College of Humanities and Sciences of Shanxi Normal University, Linfen, 041000 Shanxi, China

²Graduate School, Adamson University, Manila 0900, Philippines

³Shanxi Normal University, Shanxi 041004, China

Correspondence should be addressed to Yuanyuan Wu; 703480@sxnu.edu.cn

Received 10 May 2022; Revised 8 July 2022; Accepted 28 July 2022; Published 28 September 2022

Academic Editor: Chia-Huei Wu

Copyright © 2022 Chi Zhang and Yuanyuan Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of society, basketball has been deeply loved by people all over the world. Meanwhile, basketball has become one of the most watched events in the world. In order to adapt to the development of basketball and make people better understand the laws of basketball, this paper aims to study the application of embedded intelligent object detection system in basketball sports test and use the embedded intelligent object detection system to separate basketball sports. Compared with the general computer processing system, the embedded system has great differences. It cannot realize the large-capacity storage function because there is no matching large-capacity medium. Strengthen the learning of basketball theoretical knowledge through learning rules. Firstly, the research background of basketball and the research significance of moving target detection are introduced, and the embedded Linux operating system is introduced, including hardware design and software design. Then, four moving target detection algorithms are proposed. Object detection, also called object extraction, is an image segmentation based on the geometric and statistical features of objects, and its accuracy and real-time performance are an important capability of the entire system. Experiments show that when using the YCrCb domain detection algorithm, the average time-consuming is 292 ms, and the conversion frame rate is about 3.5fps. The YCrCb domain detection algorithm has good detection effect and high efficiency.

1. Introduction

1.1. Background. Basketball is known as “the second largest ball game in the world,” which shows that basketball has been recognized and widely participated in by everyone. Since the establishment of the Chinese Basketball Association, there have also been some players with good technical level and physical fitness, but the results of the competition are not as good as those of badminton, table tennis, and other net separation projects. To ensure the robustness of embedded systems, motion detection algorithms must be accurate and efficient. Therefore, in order to apply it to intelligent systems, we should prioritize high computational efficiency for real-time processing. Robustness is the robustness of the system, and it is the key to the survival of the system in abnormal and dangerous situations. For example, in the

case of intentional attack of computer software, whether it cannot crash or crash is the embodiment of robustness. The derived target is required to be as complete and accurate as possible, which is an error-free motion detection algorithm.

1.2. Significance. With the continuous development of modern science and technology, computer vision technology has become an important technology to replace traditional vision in the information age. It has very important applications in many fields. In the field of video monitoring of life security, the monitoring system can automatically find the target, track and take pictures of it, and trigger the alarm to realize unattended. In medicine, we can analyze the cells in exercise. In industry, it can be used for pipeline monitoring and robot action response to moving targets. In military,

it can lock and track the targets found by radar, aircraft navigation, bomb guidance, and monitoring of border crossing behavior on the border.

1.3. Related Work. With the development of society and the progress of science, more and more people study embedded system. Among them, Soyata outlines passive radio frequency (RF) energy receiving and power collection circuits for isolated communication and computing systems that cannot access the main power supply. This paper provides a unified understanding of alternative energy collection schemes and then studies RF energy collection in embedded system environment in detail. The RF technology from directional communication signal reception to decentralized environmental power collection is discussed in detail. The principle is that the scanner transmits radio wave energy of a specific frequency to the receiver, which is used to drive the receiver circuit to send out the internal code, and the scanner will receive the code at this time. A comparative focus on design tradeoffs and process changes is provided to represent the diversity of applications requiring radio frequency acquisition units. However, his research is not very practical [1]. Energy collection is a promising technology that can overcome the limitations of energy availability and prolong the service life of battery powered embedded systems. Among them, Xue studies how to prolong the service life of real-time embedded system (RTES-EH) with energy collection capability. RTES-EH includes photovoltaic (PV) panels for energy collection, supercapacitors for energy storage, and real-time sensor nodes as embedded load devices. The global controller simultaneously performs the optimal operating point tracking of PV panel, the state of charge (SOC) management of supercapacitor, and the energy collection of sensor dynamic voltage and frequency scaling (DVFS); senses the real-time task scheduling node; and adopts an accurate solar irradiance prediction method. The controller adopts cascade feedback control structure, in which the external supervisory control loop uses DVFS to perform real-time task scheduling in the sensor node, while maintaining the optimal supercapacitor SOC to improve the system availability. However, at present, it is difficult to realize all aspects [2]. Learning the design, simulation, and implementation of embedded systems opens a new paradigm for the development of practical laboratory experiments based on embedded systems. Ajao demonstrated the use of simulation computer-aided design tools such as proteus virtual system modeling (PVSM), Multisim, and micro-CAD learning system circuit design methods. In order to demonstrate how to conduct circuit experiments based on a virtual microcontroller in the laboratory, the PIC16F887 chip is used as the main logic element. Several actual hands-on experiments were demonstrated with the help of PVSM, and the results of each experiment were shown with snapshots. The purpose is to propose a learning technology method to allow students to participate in practice and become experts in learning embedded systems courses. In addition to student classroom teaching, this practical experiment will be an additional advantage and allow amateurs, professional scientists, and engineers to design

and analyze [3]. Embedded systems must resolve many potential conflicting design constraints, such as flexibility, energy, heat, cost, performance, and safety, all of which face highly dynamic operating behaviors and environmental conditions. Dutt proposed that by adding intelligent elements, we hope that the resulting “smart” embedded system can operate correctly under the expected constraints, despite the highly dynamic changes in the application and environment, as well as the underlying software/hardware platform. Since terms related to “intelligence” (for example, self-awareness, adaptability, and autonomy) are widely used in many software and hardware computing environments, a taxonomy of “self-x” terms has been proposed and used. This classification is used to correlate the main “smart” software and hardware computing tasks. One of the main attributes of the intelligent embedded system is the concept of self-awareness, which enables the embedded system to monitor its own state and behavior as well as the external environment in order to adapt intelligently. Home appliances will be the largest application field of intelligent embedded systems, and intelligent refrigerators and air conditioners will lead people’s lives to a brand-new space. Even when you are not at home, you can remotely control your home appliances through the network [4]. Many emerging applications such as the Internet of Things, wearable devices, and sensor networks have ultra-low power consumption requirements. At the same time, for cost and programmability considerations, many of these applications will be powered by general-purpose embedded microprocessors and microcontrollers (rather than ASIC). Cherupalli has taken advantage of a new opportunity to improve the energy efficiency of ultra-low-power processors, which are expected to promote these applications-dynamic timing slack. When the embedded software application executing on the processor does not execute the static critical path of the processor, dynamic timing slack occurs. In this case, the longest path executed by the application has an additional time delay. By reducing the voltage of the processor at the same frequency until the longest path executed just meets the time limit, it can be used without reducing performance costs. This time delay is used to save power. It is possible to safely allow paths that cannot be executed by the application to violate time constraints. The results show that dynamic timing slack exists in many ultra-low power applications, and the use of dynamic timing slack can bring significant power savings to any ultra-low power process. But the cost of application development is not low [5]. In recent years, technicians have continuously deepened their exploration of embedded systems. Computer vision is an interdisciplinary object detection field. Computer vision is the use of computers to process images to obtain the information we want. The target detection relay is an important part of auxiliary monitoring, vehicle detection, and attitude estimation. In this work, Madasamy proposed a new “deep YOLO V3” method to detect multiple objects. This method looks at the entire frame during the training and testing phases. It follows regression-based techniques and uses probabilistic models to locate objects. In the article, Madasamy constructed 106 convolutional layers, followed by 2 fully

connected layers and an input size of $812 \times 812 \times 3$ to detect small-sized drones. The pretrained convolutional layer is classified at half the resolution, and then the resolution is doubled for detection. The number of filters per layer will be set to 16. The number of filters in the last scale layer exceeds 16 to improve small target detection. This configuration uses an upsampling technique to improve the unwanted spectral image into an existing signal and rescale the feature at a specific location. It clearly reveals upsampling to detect small objects [6].

1.4. Main Structure. In Chapter 1, it mainly introduces the research background and significance of basketball and moving target detection and tracking, the development status and development trend of domestic and foreign theories and applications, and the main content of related work and papers. Chapter 2 introduces the software and hardware structure of embedded systems; outlines three common target detection algorithms; including the complexity and advantages and disadvantages of the algorithms; and introduces the improved four-frame hybrid differential method to detect targets and to move the implementation of four algorithms for target detection. Chapter 3 introduces the work related to experiments, Chapter 4 comparison and performance test of various algorithms after the experiment, and Chapter 5 the summary, where it summarizes the design of the entire system and the achieved target effects, evaluates the system's deficiencies, and clarifies the future development direction.

2. Basketball Mobile Test Method Based on Embedded Target Detection Intelligent System

2.1. Embedded System. Embedded system is an application-centric, computer-based dedicated computer system. Software and equipment can be reduced, and it can meet the strict requirements of the implementation system in terms of operation, reliability, cost, volume and energy consumption, etc. The embedded system is highly specialized and is usually oriented to a specific application; the miniaturization of the volume is conducive to the realization of miniaturization, and it is convenient to embed the embedded system into the target system; the real-time performance is good. As shown in Figure 1, this is an embedded system architecture based on ARM [7].

2.1.1. Hardware System Platform. In order to save time and reduce investment costs, it is very important to choose a set of integrated development environment as the development environment of ARM so that when in use the PC can support editing, compiling, assembling, linking, and other tasks. The ARM processor itself is a bit design, but it is also equipped with a 16-bit instruction set, which retains all the advantages of a 32-bit system; the addition of the DSP instruction set to the CPU function provides enhanced 16-bit and 32-bit arithmetic operation capabilities, improving performance and flexibility. What is used here is the embedded development platform with S3C214 microproces-

sor as the core of the system. As shown in Figure 2, this is the ARM program development process [8].

For the research of application platform, this system intends to adopt embedded platform as the operating environment, because compared with Windows platform, embedded platform has the characteristics of low power consumption, stability, reliability, and mobility. There are several issues that need to be addressed when developing on an embedded platform. First of all, on how to collect and store images, the technical solution to be adopted in this article is to build an ARM microprocessor system, plus appropriate peripheral equipment, to capture images through a camera connected to the ARM board. As for the storage of images, consider using SQLite, a unique database for embedded systems, because the database is syntactically similar to a relational database, which can be quickly mastered by those who have some experience in using databases. Next, we need to solve the tracking problem of moving objects in the frame sequence, and we plan to use the OpenCV open source library to design the tracking algorithm. The Opencv library is an open source library with mature computing vision. It is a cross-platform library that can be easily ported to the ARM panel without modification in the code. Finally, for the development platform of the system, it is planned to use Qt Creator. Qt is a commonly used graphical user interface development platform under Linux with abundant relevant technical documents [9].

2.2. Target Detection. Moving target detection is to extract the moving foreground from the background image. Firstly, the background model is obtained by the statistical method, and the morphological method and the area of the connected domain are used for post-processing to eliminate the influence of noise and background disturbance and obtain an accurate moving target. But often due to interference from external factors such as weather and light, motion detection is not a simple matter. Commonly used moving target detection methods include optical flow method, background difference method, and adjacent frame difference method.

2.2.1. Optical Flow Method. Optical flow refers to the brightness information of an image. The advantage of this method is that the target object can be extracted without knowing the historical data of the moving object in the image and the effect is good. The disadvantage is that the amount of calculation is too large, and the program implementation is complicated [10].

The background difference method is a common method to identify and extract moving regions. The basic principle is to subtract the current frame in the image sequence and the acquired background reference model and calculate the pixel difference with the background image, so as to determine the position, outline, size, and other characteristics of the moving object. The creation of the background model is an important step to determine the effect of motion area detection. The creation of the background model is divided into a simple background model and a Gaussian background model. The so-called simple background model uses a still

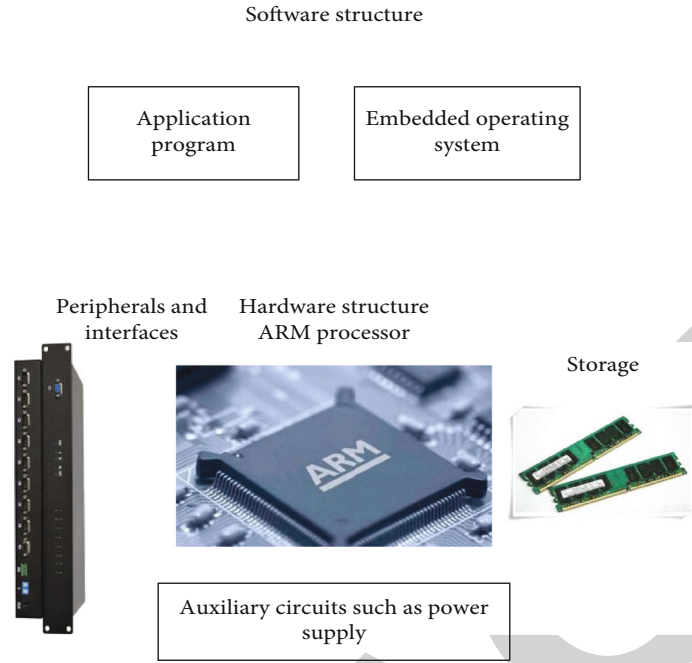


FIGURE 1: Embedded system structure based on ARM.

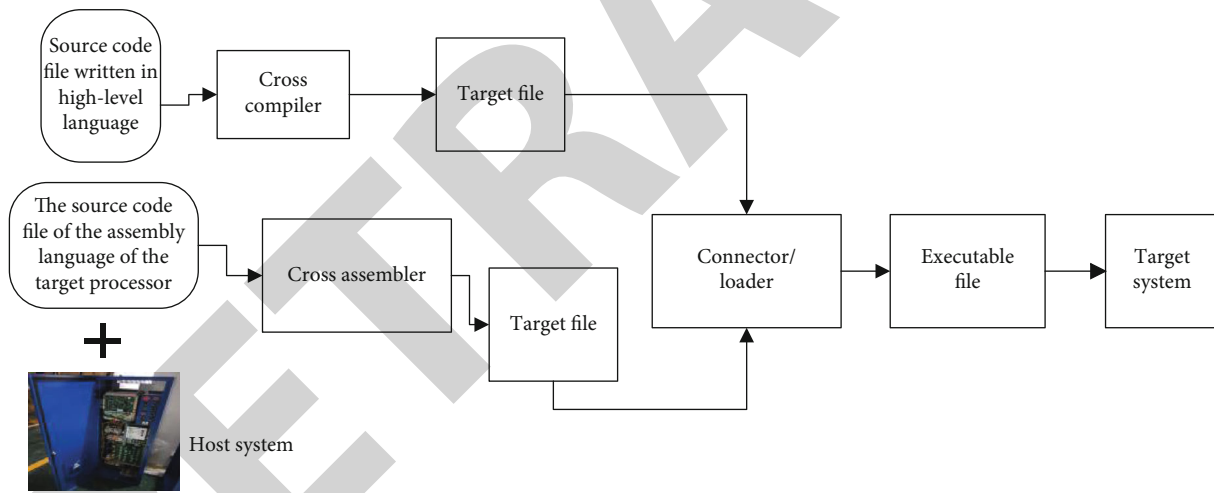


FIGURE 2: ARM program development process.

image as a background model. If the difference between the color value of the pixel at the same position in the background image and the color value of the pixel in the current frame is greater than the threshold, it is regarded as a moving target image; otherwise, it is regarded as a background image.

This algorithm is relatively simple to operate, but it is easy to affect the effect due to weather factors such as brightness, because the selected limit value is obtained through a series of experiments and is not suitable for the dynamic changes of basketball. The mixed Gaussian model uses the weighted sum of multiple Gaussian components to obtain the approximate distribution of light points, has a good detection effect, improves the shortcomings of the simple background model, and can adapt to changes in background

illumination. The disadvantage is that the amount of calculation is very large, so it will be very slow in the experiment [11].

The principle of the adjacent frame difference method is to obtain the moving target area by making the difference between two adjacent frames. This algorithm is relatively easy and can adapt to changes in brightness and other natural environments. However, when this method detects high-speed moving targets, the output area will be much larger than the actual size of the target, and it is difficult to accurately identify the slow moving and stationary objects we perceive. When facing objects with relatively uniform gray values inside, it is not possible to extract all the relevant feature pixels, and the extracted image may have a hole effect, which will affect the next experiment [12]. The concept of

the four-frame hybrid differential method is introduced below.

(1) Flow chart of the algorithm

The overall flow chart of the algorithm is shown in Figure 3.

(2) Improvement of inter-frame difference method-hybrid difference method

In this method, the first three image frames are stored, and the fourth frame image and the stored image are correspondingly differentiated two by two so that the sensitivity of the first stage recognition of moving targets can be enhanced. Let $Y_t(m,n)$ be the image at time t , $Y_{t-k}(m,n)$ be the image saved at time $t-k$, and (m,n) be the position of the same pixel in each frame of image; $A_{t-k}^t(m,n)$ represents the difference between the t and $t-k$ frames at (m, n) pixels. In continuous experimentation, it is concluded that in using the hybrid difference method to obtain object images, when $k=3$ is the most accurate and clear.

Which is

$$A_{t-3}^t(m, n) = |Y_t(m, n) - Y_{t-3}(m, n)|, \quad (1)$$

$$A_{t-2}^t(m, n) = |Y_t(m, n) - Y_{t-2}(m, n)|, \quad (2)$$

$$A_{t-1}^t(m, n) = |Y_t(m, n) - Y_{t-1}(m, n)|. \quad (3)$$

The difference image obtained by Formulas (1), (2), and (3), using threshold value discrimination, obtains the binarized image:

$$A_{t-k}^t(m, n) = \begin{cases} 1 & A_{t-k}^t(m, n) \geq T \\ 0 & A_{t-k}^t(m, n) < T \end{cases}. \quad (4)$$

Perform logical OR operation on the binarized image obtained in Formula (4), and the result obtained is the final image after mixing and difference; we get

$$A(m, n) = A_{t-1}^t(m, n) \vee A_{t-2}^t(m, n) \vee \dots \vee D_{t-k}^t(m, n). \quad (5)$$

The $A(m, n)$ obtained by Formula (5) is the final extracted and detected moving target [13].

(3) Determination of the threshold of the hybrid difference method

In this article, the threshold S is determined by an adaptive method. Choosing an accurate threshold is very important for the foreground and background of the entire segmented image. The most commonly used methods to determine the threshold are iterative method, histogram method, adaptive local threshold method, and maximum between-class variance method.

The iterative method is easier to understand, but it has a large amount of calculation and slow processing speed. The histogram method is only suitable for the phenomenon

where the image presents an obvious double-humped gray histogram. It can display the state of quality fluctuation and transmit information about the quality of the process more intuitively; after studying the state of quality fluctuation, you can grasp the state of the process and carry out quality improvement work. The adaptive local threshold method is suitable for scenes with brightness, and when there is sudden noise in the segmented image, the processing speed is also very slow, and artifacts may appear [14].

OTSU is a method of maximizing the variance between classes. It uses the most accurate boundary value to divide the gray histogram into two parts and maximizes the difference between the variance values between the two parts, that is, the maximum separation. It divides the image into two parts: background and foreground according to the grayscale characteristics of the image. The greater the interclass variance between the background and the foreground, the greater the difference between the two parts that make up the image. The background is wrongly divided into the foreground, which will cause the difference between the two parts to become smaller. This method has fast calculation speed and strong real-time performance [15]. The steps are as follows: Assuming that the gray level of the image is in the region (0~255), the gray level m has n_m pixels, and then the probability function is extracted from the gray level histogram:

$$P_m = \frac{n_m}{N}. \quad (6)$$

The threshold S is selected to divide the gray levels into two categories: $B_1 = (0, 1, 2, \dots, S)$ and $B_2 = (S+1, S+2, \dots, 255)$; therefore:

$$w_0 = p_r(C_1) = \sum_{m=0}^S p_m = w(t), \quad (7)$$

$$w_1 = p_r(C_2) = \sum_{m=S+1}^{255} p_m = 1 - w(t), \quad (8)$$

$$\mu_0 = \sum_{m=0}^S m \frac{p_m}{w_0} = \frac{\mu_t}{w_t}, \quad (9)$$

$$\mu_1 = \sum_{m=S+1}^{255} m \frac{p_m}{w_1} = \frac{\mu_T - \mu_t}{1 - w_t}. \quad (10)$$

From Formulas (7), (8), (9), and (10), we can get

$$\mu_S = \omega_0 \mu_0 + \omega_1 \mu_1, \quad \omega_0 + \omega_1 = 1. \quad (11)$$

Among them, ω_0 is the probability attributed to the foreground pixel, μ_0 is the average gray level, ω_1 is defined as the probability attributed to the background pixel, μ_1 is the average gray level, and μ_S is the average gray level of the entire frame of image. The between-class variance of C_1

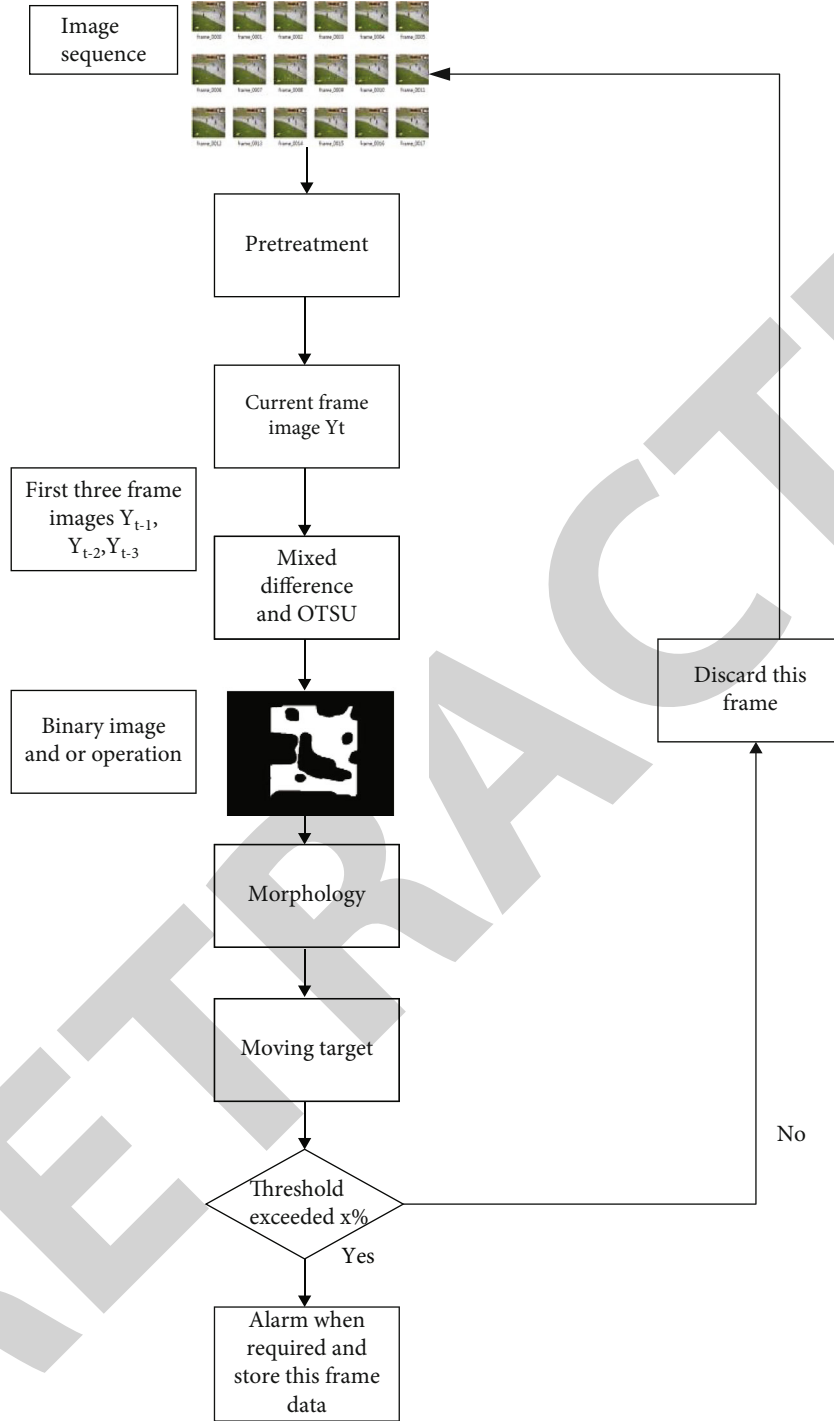


FIGURE 3: Algorithm flow chart

and C_2 is

$$\theta = \omega_0(\mu_0 - \mu_S)^2 + \omega_1(\mu_1 - \mu_S)^2. \quad (12)$$

Substitute (11) into (12) to get

$$\theta = \omega_0\omega_1(\mu_0 - \mu_1)^2. \quad (13)$$

In order to maximize the value of the variance between classes, the most appropriate threshold S is selected, which is the final threshold.

Table 1 shows the calculated thresholds for some specific frames, as follows.

In this paper, the maximum between-class variance method is improved. The improved algorithm first divides the difference image into multiple regions and then uses the algorithm to calculate the threshold for each region

TABLE 1: Thresholds calculated for different frame sequences.

Frame sequence number	OTSU threshold	Frame sequence number	OTSU threshold
100	19	120	19
150	10	180	10
200	7	230	7
250	15	280	15
300	12	330	12
320	17	350	17

and then adds all the thresholds, divides by the number of divided blocks, and calculates the average Threshold, this value is the threshold for binarization of the entire image [16]. The mathematical expression is as follows:

$$S = \sum_{m=0}^n s(m)/n. \quad (14)$$

In the formula, S represents the output result, $s(m)$ represents the threshold of the m -th small area of the image, and n represents the number of divided areas.

This is the threshold of some specific frames calculated after improvement, as shown in Table 2.

The derivation process of the OTSU method shows that the final threshold is obtained by performing variance calculation on each gray value. Assuming that the time consumption for variance calculation for each gray level is T_1 , it takes $256 T_1$ time for the entire frame of image analyze the large amount of calculations on the whole, so appropriate improvements should be made in this regard to reduce the search time and achieve high calculation efficiency.

The gray value of the image is between 0 and 255, so we choose to calculate every 10 gray levels to reduce the number of calculations and calculate the value that maximizes the between-class variance θ and set it to t and then calculate each level of gray in the interval $[t-\Delta x, t+\Delta x]$ to find the value of t that maximizes θ . There are two special cases for the required t :

- (1) $t - \Delta x < 0$; then calculate each gray level in the interval $[0, t - \Delta x]$.
- (2) $t + \Delta x > 255$; calculate each gray level on $[t + \Delta x, 255]$.

Using this method to calculate the threshold, the consumption time is $35T_1$ or $45T_1$, which is 5 to 6 times faster than the time required without improvement.

2.3. Basketball Moving Test Method Based on Target Detection

2.3.1. System Composition. This system is mainly composed of a camera, an ARM development board, and a PC-side serial port auxiliary tool, as shown in Figure 4.

The following describes the functions and characteristics of each part shown in Figure 4:

TABLE 2: Thresholds calculated by different frame sequences after improvement.

Frame sequence number	OTSU threshold	Frame sequence number	OTSU threshold
100	20	120	12
150	9	180	8
200	6	230	7
250	15	280	15
300	13	330	13
320	13	350	17

- (1) The camera used in this topic supports HTTP service, with JPEG output function, and its output image size is 320×240 ; it supports RTSP protocol and has MPEG4 video output function. This camera is the source of visual data for this system
- (2) The ARM development board CPU used in this subject is ARM S3C2410, with a main frequency of 203MHz, with one RJ-45 interface and two RS-232 interfaces. Embedded Linux is installed, kernel version 2.6.10. The main algorithm logic of this control system runs on this development board
- (3) The PC serial port auxiliary tool is a program running on the PC, which can receive, interpret, and display the serial data sent by the development board to the PC in real time and make some prompts [17].

2.3.2. Design of System Software. The main control program here is an image acquisition, decoding, filtering, segmentation, and follow-up running on the ARM development board.

In the trace multithreaded program, the main modules include image acquisition module, image decoding module, moving object detection and shadow elimination module, and moving object tracking module.

Since the target system is an embedded software system that can run on the development board in quasi-real time, its operating environment is the embedded Linux operating system, which means that the development process is roughly divided into three steps:

- (1) Implement various candidate algorithms on the PC to evaluate the effect and time and space complexity
- (2) Porting from other platforms to Linux operating system and running on board
- (3) According to the characteristics of the CPU, the key parts of the code are optimized at the C language level so that the running speed is improved and the quasi-real-time operation is realized.

If the Linux operating system has been used on the PC from the beginning of development, step one and step two can be combined. Here, for convenience in the development process, step one is completed using VC6 on the Windows platform. At this time, you can easily observe the operating

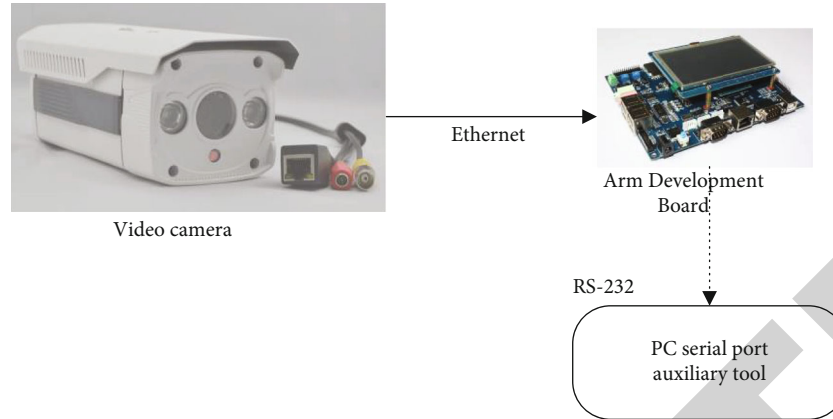


FIGURE 4: Composition of moving target detection system.

effect of the algorithm through the display module for debugging and make a subjective evaluation.

Step two is completed on the Fedora Core4 release version Linux, which is mainly due to the fact that its kernel is close to the kernel used by the embedded Linux on the development board. This step uses samba, vnc, nfs, and other services. Samba is used to conveniently access the program code on Fedora on windows and to modify it with a convenient text editing tool. vnc is a universal remote desktop program that provides a way to access Fedora graphical interface on windows for compiling, debugging, and other operations. In this way, the entire development can be completed in front of a computer, which improves development efficiency. nfs is used to enable the development board to access and run cross-compiled executable programs without programming, avoiding the slow programming process [18]. The third step is to use the ARM integrated development environment to complete on the windows platform by establishing a unit project.

2.3.3. Image Acquisition Module. The camera we use can output compressed images in JPEG format via HTTP protocol. It can also output the MPEG4 format baseline video stream through the RTP protocol, the frame rate is 25 frames per second, and the I frame period is 30 frames.

The reason why MPEG4 video stream is not used as the source of visual information is mainly due to the following considerations. Because the MPEG4 format baseline video stream essentially uses the H.263 protocol processing method, its P frame decoding complexity is relatively small, but the P frame decoding complexity is relatively large; if the H.263 protocol is used as the visual data source, it will cause a lot of frame loss due to the inability to process in time. If the P frame is lost, the decoding will fail, and the picture cannot be recomposed until the I frame appears. The I frame period is 30 frames and will only appear after about 1.2 seconds, which will cause serious real-time problems. Therefore, compressed images in JPEG format are used as visual data sources [19].

The image acquisition module is responsible for using HTTP protocol to obtain JPEG format picture files from the web camera. Since the network is an unstable and unreliable environment, you must make full legality judgments

before doing JPEG decoding. This can reduce the impact of the system from network equipment errors and improve the robustness of the system.

The basic steps are as follows:

- (1) Establish a TCP connection with port 80 of the camera
- (2) Send HTTP GET request
- (3) Receive the response, and judge whether the response code is OK; if not, return
- (4) Extract the content and make some judgments to prevent the occurrence of errors
- (5) If it is normal, the JPEG image is obtained, and the starting address of the image is returned to [20]

2.3.4. Implementation and Optimization of Image Decoding Module. JPEG image file decoding steps are roughly as follows:

- (1) Read the file information from the file header. JPEG file data is divided into two parts: file header and image data. The file header records important information, such as image output, length and width, sampling factor, quantum table, and Huffman table. Therefore, before decoding, for image data decoding processing, the information in the file header should be read
- (2) Read the MCU from the image data stream, and extract each internal color unit on how to separate continuously stored MCUs from the data stream and how to separate multiple color elements from each MCU
- (3) The color component unit shall be restored from the data stream to the matrix data. Use the Huffman table provided in the file header to decode the component unit and return it to the 8×8 data matrix

The 8×8 data sheet should be further decoded. This part of the decryption work takes 8×8 data units as a unit and

includes five steps: differential decoding of the direct current coefficients of adjacent matrices, reverse quantum data using the quantum table provided in the file header, reverse

zig-zag glyph encoding, interlaced positive and negative correction, and reverse discrete synthesis. The final output is still in the form of a data matrix 8×8 .

- (4) The color system YCrCb is converted to RGB. Integrate the decoding results of each color unit of the MCU to convert the image color system from YCrCb to RGB
- (5) Classification and sorting of decoded data for each MCU. Continue to read and decode the MCU in the data stream until all MCU are read, and the decoded data of each MCU is correctly organized in the complete picture [21]

2.3.5. Realize the Moving Object Detection Algorithm. Moving object detection is an important technology used here. Moving object detection is one of the basic functions of image content analysis. It is usually a process of generating an alarm when it is confirmed that the suspicious target is indeed a foreign moving individual after a certain period of time. The currently implemented moving object detection algorithms have three simple background subtraction methods, single Gaussian background model method, and mixed Gaussian background model method. The principles and characteristics of the three algorithms are compared as follows:

- (i) Simple background subtraction: Use a fixed frame as the scene, subtract the current frame from the background frame, and use the consistent threshold to determine whether the pixel is background. All calculations are done in the gray domain. The advantage is low time and space complexity, but due to the use of a unified threshold, the effect is moderate.
- (ii) Single Gaussian model method: Each pixel uses a Gaussian model as the background point type and matches the current frame pixel with the background point model. If it falls within the model, it is judged as the background, and the model mean and variance are updated. On the contrary, it is judged as the foreground and background learning is not carried out. All calculations are done in the gray domain. The effect is moderate, the time complexity is greater than the background subtraction method, and the space complexity is greater [22].
- (iii) Mixed Gaussian model method: Each pixel uses multiple Gaussian models as the background point model and uses the weight to represent the possibility that the model is a background. The model with the largest weight is considered the background, and the remaining models are considered the foreground. After each pixel is matched, the weight, model mean, and variance are updated once. All

calculations are done in the gray domain. The adaptive ability is strong, and the effect is better, but the time and space complexity are large [23]. A mixture model is a probabilistic model that can be used to represent K subdistributions in the overall distribution. It does not require the observation data to provide information about the subdistribution, but only calculates the probability of the observation data in the overall distribution.

The principle of the shadow elimination part is as follows:

When a point is judged as a front scenic spot by the above detection algorithm, the change in the HSV domain of the point relative to the background point is detected. It is removed from the foreground.

This module will finally obtain the binary result of the pixel domain separated from the front and background, the static background is represented by 0, and the moving foreground is represented by 1 [24].

Since the YCrCb domain itself also directly characterizes the luminance and chrominance information, the work of the HSV domain should be replaced by the YCrCb domain to a certain extent. Since the common shadow elimination module considers that the chroma and saturation changes less and the brightness changes a lot, it is a shadow, so it can also be considered to directly use a large chroma change as the basis for foreground detection. Purely changing from subtraction logic to supplementary logic, even if noise may be introduced, these noises consist of two parts: one is true shadows that should be subtracted but not subtracted, and the other is false prospects that should not be supplemented but added. The large chromaticity change has prevented the false foreground from being supplemented. The only problem is the introduction of true shadows, which can be offset by increasing the brightness change threshold [25].

Therefore, a prospect admission rule characterizing the YCrCb domain is designed here, which is briefly described as follows.

Let Y_m , Cr_m , Cb_m , and N represent the Y, Cr, and Cb component values of a certain pixel in the current frame. Y_{mn} , Cr_{mn} , Cb_{mn} , respectively, represent the values of Y, Cr, and Cb components of the corresponding pixels of the background frame.

Delimit:

$$\begin{aligned}\Delta Y &= |Y_m - Y_{mn}|, \\ \Delta Cr &= |Cr_m - Cr_{mn}|, \\ \Delta Cb &= |Cb_m - Cb_{mn}|, \\ \Delta rb &= |\Delta Cr - \Delta Cb|,\end{aligned}\tag{15}$$

$$R_m = \begin{cases} 1 & \text{Foreground } \Delta Y \geq T_Y \cup \Delta rb \geq T_{rb} \\ 0 & \text{Background otherwise} \end{cases}$$

2.3.6. Realization of Moving Object Prediction Algorithm. In this paper, Kalman filter is used to predict the area of the moving object in the next frame, and the predicted area is

TABLE 3: Basic information of tennis players.

Name	Xiao Ling	Li fen	Average
Age	26	27	26.5
Height (cm)	175	176	175.5
Weight (kg)	69	69	69
Year of exercise (years)	4	4	4
Clapping hands	Right hand	Right hand	—
Backhand style	Both hands backhand	Both hands backhand	—

obtained. In order to improve the real-time performance of the algorithm, a simple constant velocity model is used. Suppose the target position is $x_p(n)$ 、 $x_q(n)$, the speed is $v_p(n)$ 、 $v_q(n)$, and the frame interval is T (when tracking in real time, T is 1), assuming $u_1(n)$ 、 $u_2(n)$ is zero mean variance, corresponding to Gaussian white noise of σ_1^2 、 σ_2^2 , respectively, and $E[u_1(n)u_2(n)] = 0$, then Here the equation of motion is

$$x(n+1) = Ax(n) + w(n) \quad (16)$$

Then,

$$\begin{aligned} x(n) &= [x_p(n) \ v_p(n) \ x_q(n) \ v_q(n)]^T, \\ w(n) &= [0 \ u_1(n) \ 0 \ u_2(n)]^T, \end{aligned} \quad (17)$$

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

The matching template measures the horizontal and vertical positions of the moving objects, which are marked as $a_1(n)$ and $a_2(n)$. The measured noise $v_1(n)$ 、 $v_2(n)$ is Gaussian white noise with zero mean and variance σ_p^2 、 σ_q^2 , respectively. The equation is

$$x(n+1) = Ax(n) + w(n). \quad (19)$$

In particular,

$$\begin{aligned} a(n) &= [a_1(n) \ a_2(n)]^T, \\ v(n) &= [v_1(n) \ v_2(n)]^T, \\ H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned} \quad (20)$$

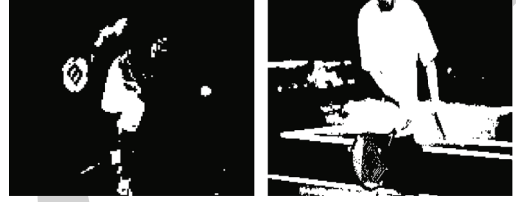


FIGURE 5: Test effect diagram of single Gaussian model detection algorithm.



FIGURE 6: Test effect diagram of YCrCb domain background subtraction model detection algorithm.

Kalman predicts the iteration process as follows:

$$\begin{aligned} B(n) &= AP(n|n-1)H^T(HP(n|n-1)H^T + C)^{-1}, \\ \hat{x}(n+1|n) &= A\hat{x}(n|n-1) + G(n)(x(n) - H\hat{x}(n|n-1)), \\ p(n) &= (I - A^{-1}G(n)H)P(n|n-1), \\ p(n+1|n) &= AP(n)A^T + D, \end{aligned} \quad (21)$$

In particular,

$$D = E[w(n)w^T(n)] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_2^2 \end{bmatrix}, \quad (22)$$

$$C = E[v(n)v^T(n)] = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_q^2 \end{bmatrix}. \quad (23)$$

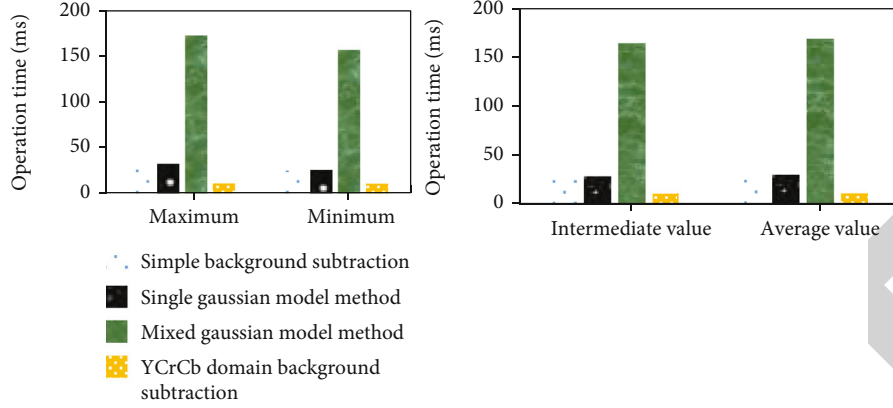


FIGURE 7: Performance test of moving object detection algorithm.

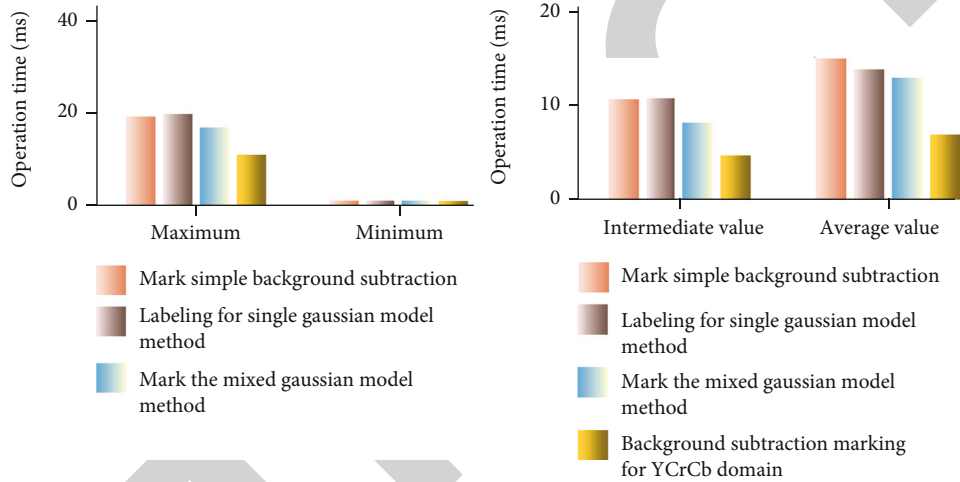


FIGURE 8: Performance test of moving object detection algorithm.

The initial conditions of Kalman iteration are $\hat{x}(1|0), P(1|0)$, namely:

$$\begin{aligned} \hat{x}(1|0) &= [a_1(2) \ [a_1(2) - a_1(1)]/T \ a_1(2) \ [a_2(2) - a_2(1)]/T]^T, \\ P(1|0) &= AP(0)A^T + D. \end{aligned} \tag{24}$$

3. Related Experiments of Basketball Movement Test Based on the Embedded Target Detection Intelligent System

3.1. *Subjects.* Because there are more basketball games on outdoor courts, the test was conducted in a mild outdoor environment with more uniform light and low wind speed. Choose an appropriate angle for easy collection of test material. The testing time is fixed from 2 pm to 5 pm to ensure that the influence of external factors is within acceptable limits. The experimental preparation stage is mainly aimed at the inspection of the material data playback material and the debugging of the equipment to ensure that no special failure occurs. Before the experiment, explain the exper-

TABLE 4: Performance test of moving object tracking algorithm.

Attributes	Index value (ms)
Maximum	0.0
Minimum	0.0
Average	0.0

imental process and precautions to the basketball players. Table 3 is the basic information of the six basketball players tested.

3.2. *Experimental Procedure.* Use the RJ-45 interface used by the ARM development board to request pictures from the network camera via Ethernet, and receive the configuration information sent by the PC management program. Note that there are two RS-232 interfaces: One is used as a console to communicate with the host PC for development, and the other is used to send moving object detection information to the PC monitoring program and to record experimental data in time [26].

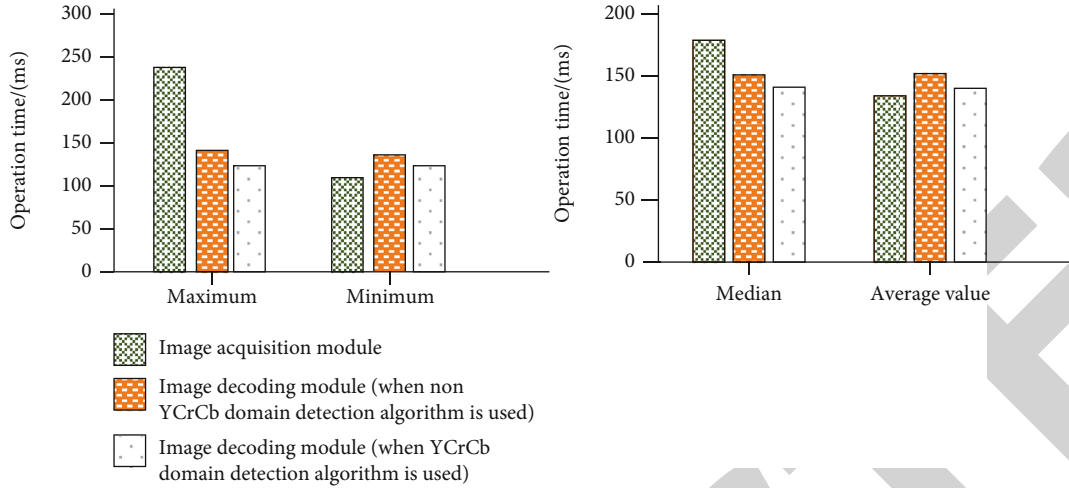


FIGURE 9: Performance test of moving object detection algorithm.

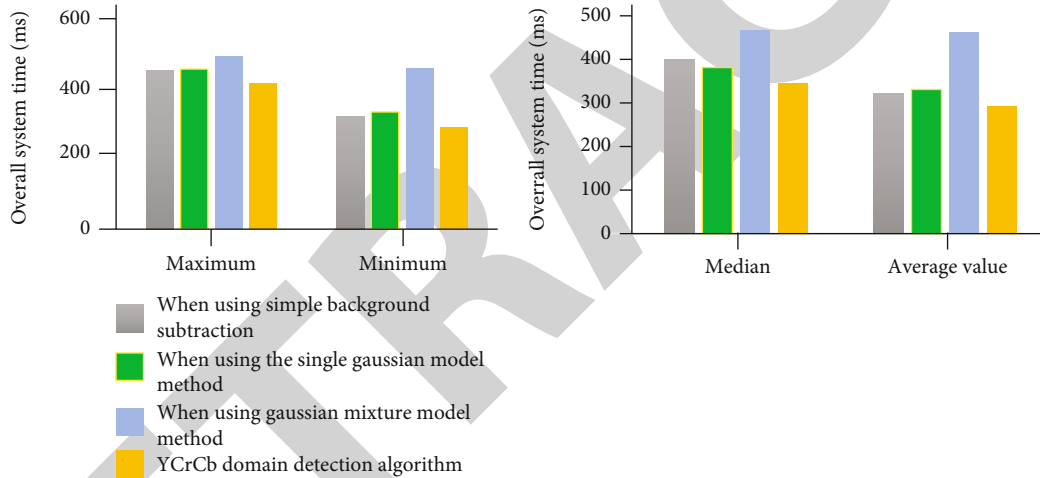


FIGURE 10: Performance test of moving object detection algorithm.

4. System Performance Test and Evaluation

4.1. *Effect and Performance Test of Moving Object Detection Algorithm.* The renderings of each algorithm are given below.

As shown in Figure 5, when the light field is relatively stable, the first three algorithms have similar effects, and all have the phenomenon of foreground holes. This is mainly because the gray values of the foreground and background are relatively close, leading to missed judgments.

As can be seen from Figure 6, the background subtraction method of YCrCb color space can better protect the integrity of the foreground in this test. However, when the front and background brightness is close and the color is similar, the phenomenon of voids will also occur. Since the information used here comes from JPEG files, there is a large quantization error, and it is easy to produce shot noise on the background.

As shown in Figure 7, the running time test of the above four algorithms on the development board is as follows. The structure of the test is obtained by time stamping and subtracting, and the test sequence length is 1000 frames.

4.2. *The Effect and Performance Test of the Connected Domain Labeling Algorithm for Binary Images.* As shown in Figure 8, the following is the runtime test of this algorithm implementation on the development board. Use the YCrCb algorithm result as the test sequence; the test sequence length is 1000 frames.

4.3. *Performance Test and Comparison of Moving Object Tracking Algorithm.* Table 4 is the running time test of this algorithm implementation on the development board. The test sequence length is 1000 frames.

Since the background of the first few frames is almost pure and no effective moving object tracking window is detected, the minimum value is 0. The maximum value is

TABLE 5: Converted to frame rate.

	Converted frame rate
When using simple background subtraction	3.0
When using the single Gaussian model method	3.2
When using the mixed Gaussian model method	2.0
YCrCb domain detection algorithm	3.5

also 0. This is mainly because the number of effective moving object tracking windows is small, the problem scale is extremely small, and the calculation complexity of the Kalman filter is fixed and extremely small, so this part of the time is extremely small [27].

4.4. Overall System Performance Evaluation. Figure 9 is the sub-item performance test of the remaining modules on the development board and the overall performance of the system. The test input at this time is the picture collected from the camera in real time.

According to the experimental results, the overall time consumption of the system can be obtained as shown in Figure 10.

Table 5 shows the converted frame rate under the 4 algorithms.

It can be seen from Figure 10 that if the YCrCb domain detection algorithm is used, the average time consumption is 292 ms. From Table 5, it can be seen that the equivalent frame rate is about 3.5fps. When using other algorithms, the frame rate is also more than 2 frames, which can basically meet the requirements of quasi-real-time processing and can be used for the detection of slow objects [28]. But the YCrCb domain detection algorithm has better detection effect and higher efficiency.

5. Conclusions

With the rapid development of embedded technology, embedded systems have been applied in many fields. This article uses knowledge in the field of embedded systems and target detection to complete the design of the system software, the preparation of the software and hardware environment, and the detection of moving targets. The design and implementation of the moving target detection system is based on traditional monitoring technology. The hardware platform uses the ARM development board with the S3C2410 processor as the core. The embedded operating system is a Linux system. The system has low power consumption and high reliability. The experiment is very applicable. The later research work can be improved from the following points: (1) The selected environment is relatively simple, and more changes should be made during the experiment time and venue and (2) how to make the detection more sensitive, optimize the algorithm, etc.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] T. Soyata, L. Copeland, and W. Heinzelman, "RF energy harvesting for embedded systems: a survey of tradeoffs and methodology," *IEEE Circuits & Systems Magazine*, vol. 16, no. 1, pp. 22–57, 2016.
- [2] L. Xue, Y. Wang, N. Chang, and M. Pedram, "Concurrent task scheduling and dynamic voltage and frequency scaling in a real-time embedded system with energy harvesting," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1890–1902, 2016.
- [3] L. A. Ajao, J. Agajo, J. G. Kolo, M. A. Adegboye, and Y. Yusuf, "Learning of embedded system design, simulation and implementation: a technical approach," *American Journal of Embedded Systems and Applications*, vol. 3, no. 3, pp. 35–42, 2016.
- [4] N. Dutt, A. Jantsch, and S. Sarma, "Toward smart embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 15, no. 2, pp. 1–27, 2016.
- [5] H. Cherupalli, R. Kumar, and J. Sartori, "Exploiting dynamic timing slack for energy efficiency in ultra-low-power embedded systems," *Computer Architecture News*, vol. 44, no. 3, pp. 671–681, 2016.
- [6] K. Madasamy, V. Shanmuganathan, V. Kandasamy, M. Y. Lee, and M. Thangadurai, "OSDDY: embedded system-based object surveillance detection system with small drone using deep YOLO," *EURASIP Journal on Image and Video Processing*, vol. 2021, no. 1, p. 14, 2021.
- [7] E. Mastinu, P. Doguet, Y. Botquin, B. Hakansson, and M. Ortiz-Catalan, "Embedded system for prosthetic control using implanted neuromuscular interfaces accessed via an osseointegrated implant," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 4, pp. 867–877, 2017.
- [8] Z. Kaleem, T. M. Yoon, and C. Lee, "Energy efficient outdoor light monitoring and control architecture using embedded system," *IEEE Embedded Systems Letters*, vol. 8, no. 1, pp. 18–21, 2016.
- [9] M. Lv, "RETRACTED ARTICLE: Agricultural climate change and multilingual GIS database translation system based on embedded database and artificial intelligence," *Arabian Journal of Geosciences*, vol. 14, no. 11, pp. 1–20, 2021.
- [10] J. Liu, Z. Xu, J. Yang, and Z. Zhang, "Modeling and analysis for global and local power flow operation rules of UPFC embedded system under typical operation conditions," *IEEE Access*, vol. 8, pp. 21728–21741, 2020.
- [11] A. Z. Fekete, A. Kelemen, and L. Jakab-Farkas, "Multilevel distributed embedded system for control of the DC magnetron sputtering process," *Nephron Clinical Practice*, vol. 9, no. 1, pp. 43–55, 2017.
- [12] K. Koennoelae, S. Suomi, T. Maekilae, T. Jokela, V. Rantala, and T. Lehtonen, "Agile methods in embedded system development: multiple-case study of three industrial cases," *Journal of Systems & Software*, vol. 118, pp. 134–150, 2016.

- [13] W. J. Cho, H. J. Kim, D. H. Jung, C. I. Kang, G. L. Choi, and J. E. Son, "An embedded system for automated hydroponic nutrient solution management," *Transactions of the ASABE*, vol. 60, no. 4, pp. 1083–1096, 2017.
- [14] T. G. Cervero, J. Caba, S. Lopez et al., "A scalable and dynamically reconfigurable FPGA-based embedded system for real-time hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol. 8, no. 6, pp. 2894–2911, 2015.
- [15] B. Wang, Y. Chen, D. Liu, and X. Peng, "An embedded intelligent system for on-line anomaly detection of unmanned aerial vehicle," *Journal of Intelligent and Fuzzy Systems*, vol. 34, no. 6, pp. 3535–3545, 2018.
- [16] L. Luo, Z. X. Yang, L. Tang, and K. Zhang, "An ELM-embedded deep learning based intelligent recognition system for computer numeric control machine tools," *IEEE Access*, vol. 8, pp. 24616–24629, 2020.
- [17] S. S. Goel, A. Goel, M. Kumar, and G. Moltó, "A review of internet of things: qualifying technologies and boundless horizon," *Journal of Reliable Intelligent Environments*, vol. 7, no. 1, pp. 23–33, 2021.
- [18] J. Kwak and J. Lee, "Minimizing capacity degradation of heterogeneous batteries in a mobile embedded system," *IEEE Embedded Systems Letters*, vol. 12, no. 1, pp. 25–28, 2020.
- [19] B. Kim and H. Yang, "Reliability optimization of real-time satellite embedded system under temperature variations," *IEEE Access*, vol. 8, pp. 224549–224564, 2020.
- [20] M. Mahbub, M. Hossain, and M. Gazi, "IoT-Cognizant cloud-assisted energy efficient embedded system for indoor intelligent lighting, air quality monitoring, and ventilation," *Internet of Things*, vol. 11, p. 100266, 2020.
- [21] A. Soetedjo and I. K. Somawirata, "Implementation of eye detection using dual camera on the embedded system," *International Journal of Innovative Computing Information and Control*, vol. 13, no. 2, pp. 397–409, 2017.
- [22] A. E. Onoja, A. M. Oluwadamilola, and L. A. Ajao, "Embedded system based radio detection and ranging (RADAR) system using Arduino and ultra-sonic sensor," *American Journal of Embedded Systems and Applications*, vol. 5, no. 1, pp. 7–12, 2017.
- [23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [24] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- [25] C. Rennie, R. Shome, K. E. Bekris, and A. F. de Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robotics & Automation Letters*, vol. 1, no. 2, pp. 1179–1185, 2016.
- [26] H. Song and M. Brandt-Pearce, "A 2-D discrete-time model of physical impairments in wavelength-division multiplexing systems," *Journal of Lightwave Technology*, vol. 30, no. 5, pp. 713–726, 2012.
- [27] M. Alshehri, "An efficient smartphone assisted indoor localization with tracking approach using glowworm swarm optimization algorithm," *Journal of Cybersecurity and Information Management*, vol. 6, no. 2, pp. 101–107, 2020.
- [28] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.