International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Optimized Motion Detection Based Security System Using ARM9 and Linux

N. Kalyani
M.Tech. Student,
nareddy.kalyani@gmail.com

A. Krishna Kumari
Prof., CVSR College of Engineering
akk_gnec@rediffmail.com

Abstract - **The Embedded motion detection system involves various aspects such as the selection of hardware platform and the embedded operating system. The development board with Samsung S3C2440 is selected as the hardware platform. Start-up codes, OS kernel and users' application programs are together stored in a NAND FLASH whose capacity is 64MB. Application programs run in 64MB SDRAM, which can also be used as the room of data and the stack.**

A CMOS camera capturing videos is connected to a USB interface in the board. After the captured video is processed by the detection module, they are transmitted to a local host through the Ethernet technology.

The design for the embedded-motion detection system involves various aspects such as the selection of the hardware platform and the embedded operating system. Besides these basic conditions, the motion detection algorithm, as the brain of the whole system, is vital yet. Once an unknown motion object invades the surveillance scene, the detection algorithm will run to analyze the captured image and estimate the unknown target. Based on that analysis, the system will make decision whether to send an Buzzer to monitoring center.

Keywords – **Motion Detection, Image Processing, Detection Algorithm, Embedded Linux.**

I. Introduction

Motion detection system is being developed towards the trend of the intelligence and the miniaturization. Image processing technology plays a key role to support motion detection. On the other hand, people have developed many portable and low-cost digital products on the development board, which is capable of processing video image. That makes it possible that motion detection system becomes smaller. As a result, the efficiency of motion detection is better and people's workloads are reduced greatly.

The design for the embedded-motion detection system involves various aspects such as the selection of the hardware platform and the embedded operating system. Besides these basic conditions, the motion detection algorithm, as the brain of the whole system, is vital yet. Once an unknown motion object invades the surveillance scene, the detection algorithm will run to analyze the captured image and estimate the unknown target. Based on that analysis, the system will make decision whether to send an alarm to monitoring center.

In this paper, we will discuss how to build up the hardware and software platform based on embedded development. To improve the systemic performance, we will introduce not only how to select the detection algorithm, but also how to optimize it. By that, the detecting effect is expected to be promoted to a certain extend.

II. Hardware and Software Platform

The development board with Samsung S3C24100A microprocessor [3] is selected as the hardware platform. CPU frequency is up to 203MHz in the board. Start-up codes, OS kernel and users' application programs are together stored in a NAND FLASH whose capacity is

64MB. Application programs run in 64MB SDRAM, which can also be used as the room of various data and the stack. A CMOS camera capturing videos is connected to a USB interface in the board. After the captured video is processed by the detection module, they are transmitted to a local host or the remote monitoring center on Internet from a 10Mbps' Ethernet interface.

The embedded Linux 2.6.12 is a kind of miniature operating system, which is designed for the demand of the embedded OS [4]. It has some advantages, such as small code amount, fast running speed, strong stability and so on. And this OS cuts out the normal Linux and becomes much smaller in size. It can even be solidified in a memory chip with a few KB or MB. The kernel of Linux2.6.12 can be customized by development engineers in terms of the actual demand. So it is regarded as the ideal software platform to develop embedded application programs.

In summary, the development platform includes the target board with the S3C24100A microprocessor and the embedded Linux2.6.12. The former constitutes hardware system architecture, which is as following Fig.1. and the latter, as embedded OS, provides powerful support for the development of application software.

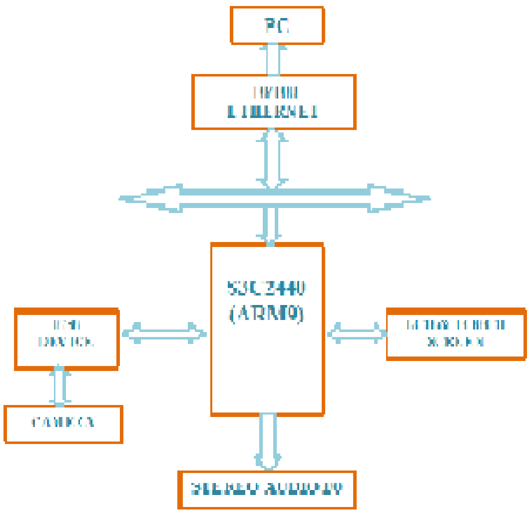


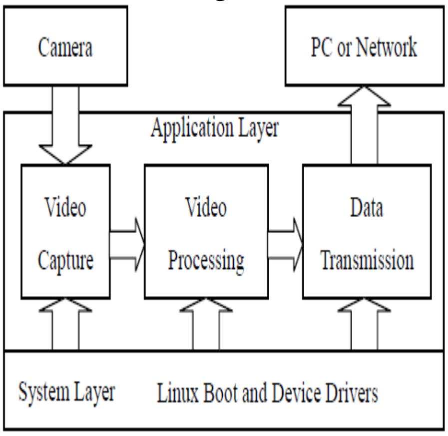
Fig.1. System Processing Framework

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

III. System Processing Frame Work

The software development mainly includes namely, Video Capture Module, Video Processing Module and Data Transmission Module. The function of every module is as following.

1. Video Capture Module. Receive video streams from the camera and transmit these bit-stream to the Video Processing Module.
2. Video Processing Module. Check up video streams and process images in video according to a detection algorithm. It is the hardcore of the entire system. When the surveillance scene is changed, it will decide to whether to send out an alarm.
3. Data Transmission Module. Output the processed result from the network interface to a local host or a monitoring center on Internet The complete processing framework for the motion detection system is shown as Fig.2.



Our software architecture is based on C/S mode. As the server, the detection system in the target board transmits video data to the client on Internet or local host. After system-layer programs is firstly loaded and starting, application-layer programs will run to capture video images by camera. Having been processed, video images(sometimes with an alarm) will be transmitted to the client where watchers can observe the monitored image The main duty of system-layer software involves start-up programs of Linux2.6.12 and low-level drivers for various devices in development board(e.g., camera, USB interface and network. SAMSUNG's S3C2440A 16/32-bit RISC microprocessor. SAMSUNG's S3C2440A is designed to provide hand-held devices and general application with low-power, high performance microcontroller solution in small die size.

IV. Detection Algorithm

A. Classic Algorithm

After many years hard work, people have presented various kind of motion object detection algorithms [5]. Every kind of algorithm is fit to a certain specific environment and has advantages under some specified conditions. By far there is not one general algorithm that is fit to all places due to the diversity and complexity of the actual situation. Among the existed classic algorithms, three kinds of algorithms are the most representative. Their titles and principles are described as following.

1. Background Subtraction Algorithm. Build up an initial model of a static scene and make the current image in video subtract the static image in initial model. As a result, the background of the image is going to be eliminated, and then the motion object is going to be segmented and detected.
2. Optical Flow Algorithm. Segment motion objects in terms of the dispersion characteristics of optical flow. The motion objects only exist in these places where the optical flow is a lot different from the other locations.
3. Image Difference Algorithm. Capture continuous images in video and have the current image subtract the next one. The places where the difference values of two images are beyond a critical number, will be regarded as motion object's location.

B. Selected Algorithm

According to the above analysis, the selected algorithm is mainly based on Image Difference Algorithm because this method is fit to the more extensive and complex situations. Assume *f(x,y,i)* and *f(x,y,j)* denote two continuous frames in video and their difference following formula In the formula(1), *t* is a threshold value that denotes a critical value, which is used to distinguish motion objects and backgrounds from an image. If the difference value of two corresponding pixels in adjacent images is no less than *t*, these pixels' values of *bidf*(*x*,*y*,*i*,*j*) are assigned backgrounds from an image. If the difference value of two to '1', which means motion object pixels of the image. In contrast, the pixels' values are assigned to '0', which means background pixels. So that the motion object can be segmented from the background in the image.

V. Algorithm Optimization

Image Difference Algorithm works well in many static or variable scenes. It has been widely applied due to less calculation and quick speed in real-time processing. But there still are some shortcomings with this algorithm. In the following situation, the algorithm may fall in malfunction and the detection result may be unreliable yet.

1. If motion objects move slowly, the changes of two Linux Boot and Device Drivers images can be very little. As a result, motion objects may not be detected.
2. In image, the segmented area (regarded as motion objects) is usually bigger than the real object's. So this may result to locate motion object in the inaccurate position.
3. The algorithm is easily interfered with the image noise, which can bring errors and affect the precision of the threshold value. Pointing to these problems, we will take the following measures to overcome these shortcomings further.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

A. Image Noise Removing and Morphology Processing

There is so much space-relativity among adjacent pixels, whereas the image noise is statistically independence. So the average gray level of pixel neighborhood can substitute the original value in noise region. This method can remove image noise effectively. Next the image is going to be processed by morphology methods. Some isolated points in the processed image are removed by the opening operation. And some small bulges in the difference image should be removed still. By the closing operation, some small holes and apertures in the difference image are going to be filled so that the image can become more smooth.

B. Accumulative Difference Image

The initial value of the accumulative difference image is assigned to '0', and the subsequent image will have some increments compared to that model. $adp(x,y,i)$ in formula(2) denotes the *i*th image frame.

By accumulative difference, we can find out whether the changes of the image gray-level obey to a rule. If the rule exists, the conclusion is that the change of gray level does not result from the noise. And those changing regions will be protected when removing image noises. This measure can avoid that the move-slowly objects are mistaken for noises and filtered wrongly because their increments in the difference image are very little.

C. Adaptive Thresholding Technique

The key factor is the threshold t, which can determine pixel attribute, namely, either motion object or background.

The optimal threshold can be figured out by the Ostu algorithm [6] based on the maximum variance criterion. But the threshold, which is obtained by Ostu algorithm, is usually a little bigger than the actual one. That may result in that small motion object should be not detected. The proportion coefficient between the motion object and the background is added to our improved method. When the coefficient is less than 0.05, the probability of the occurrence of motion objects will have been kept a constant value, namely, properly increasing motion objects' proportion in the image when their sizes become small. T he measure can avoid losing small motion objects.

VI. Experimental Results

To prove the actual effect, a lot of tests have been done for our motion detection system. The test scene is located in the street and our system will keep monitoring a certain region of the street by camera. After all drivers and application programs are loaded to the target board, video data are successfully transmitted to our host screen from the Ethernet interface in that device. The background image is initially inputted to the system model. If a motion object breaks in the surveillance region, the detection system will create the difference image, analyze the processed results and decide whether to send an alarm to the monitoring center. We intercept part of source images and difference images as following Fig.3.

VII. Conclusion

(a) Background (b) Motion Object (c) Difference Image

Fig.3. Screenshot and Difference Image

The project "Design for Motion Detection System Based on Embedded Linux**"** has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced ARM9 board and with the help of growing technology the project has been successfully implemented.

References

[1] I. Haritaoglu, D.Harwood,L.S.Davis.W4:Real-time surveillance of people and their activities[C]. IEEE Trans. Pattern Anal. Machine Intell.,Vol.22,pp.809-830,Aug.2000

[2] Huwer S. Adaptive change detection for real-time surveillance applications[M].[S.l.]:IEEE Press, 2000:37-46.

[3] CPU-S3C2410X-revl.pdf. Samsung Electronics.

[4] Daniel P. Bovet etc. Understanding The Linux Kernel [M]. O' Reilly, 2006:143-145.

[5] Mohammad Reza Javan, etc. An Efficient Object Segmentation Algorithm for Surveillance Systems[C],IEEE,2007.

[6] N. Ostu. Threshold Selection Method from Gray-Level Histograms[C].IEEE Trans. on System, Man, and Cybernetics,1979,9(1):62-66

[7] Hongzhe Han, Zhiliang Wang. Adaptive Background Modeling with Shadow Suppression[C], IEEE, 2003.