# Low Cost Real-Time System Monitoring Using Raspberry Pi

Huu-Quoc Nguyen, Ton Thi Kim Loan, Bui Dinh Mao and Eui-Nam Huh
Computer Engineering Department
Kyung Hee University
Yongin, South Korea
Email: {quoc, loanttk, mao.bui, johnhuh}@khu.ac.kr

***Abstract*—Nowadays, the Closed-Circuit Television (CCTV) surveillance system is being utilized in order to keep peace and provide security to people. There are several defects in the video surveillance system, such as: picture is indistinct, anomalies cannot be identified automatically, a lot of storage spaces are needed to save the surveillance information, and prices remain relatively high. This paper describes the design and implementation of a low-cost system monitoring based on Raspberry Pi, a single board computer which follows Motion Detection algorithm written in Python as a default programming environment. In addition, the system uses the motion detection algorithm to significantly decrease storage usage and save investment costs. The algorithm for motion detection is being implemented on Raspberry Pi, which enables live streaming camera along with detection of motion. The live video camera can be viewed from any web browser, even from mobile in real-time.**

***Keywords*—*Video surveillance; Raspberry Pi; Motion Detection; streaming***

## I. Introduction

Closed-circuit television monitoring system has now become an indispensable device in todays society [1]. Supermarkets, factories, hospitals, hotels, schools, and companies are having their own CCTV system for 24/7 monitoring. Instead of using the traditional wireless CCTV surveillance cameras, customers can now own their inexpensive security systems with the tiny super computer called Raspberry Pi [2]. IP cameras can serve better as they can send and receive data via computer network and internet based on internet protocol. Also resolution clarity of IP cameras is far better than CCTV cameras. A camera module connected to the Raspberry Pi will records all the happenings in the monitored area and live streaming can be viewed from any web browser, even from mobile in real-time.

There are many problems in the video surveillance system [3], such as: picture is indistinct, anomalies cannot be identified automatically and a lot of storage spaces are needed to save the surveillance information. Moreover, in recent years, Motion Detection [4], [5] has attracted a great interest from computer vision researchers due to its promising applications in many areas, such as video surveillance, traffic monitoring or sign language recognition. To overcome storage spaces issue, we apply the Motion Detection algorithm for live camera streaming, this allows the system to analyze incoming images from cameras, and recognize when movement occurs. And then, the video system can collect and store the most importance items for the administrators review. Effective use of Motion Detection promotes an automated approach to video system reaction, and does not require an individual to start and stop recording sessions. So we can significantly decrease storage usage and save investment costs.

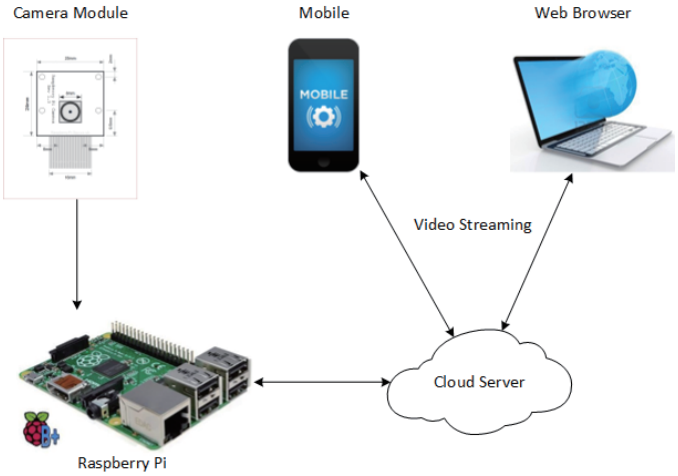



Fig. 1: System Architecture

The rest of this paper is organized as follows. In Section II presents the system architecture. Section III presents the Motion Detection algorithm. The experiments and results are demonstrated in Section IV. Finally, conclusions and future works are given in Section V.

## II. System Architecture

The system architecture is shown in Figure 1. The camera module connected to the Raspberry Pi board can be used to take high definition video, as well as stills photographs.The setup Python script will automatically deliver video data streaming to cloud server. Users will be able to watch the video stream from cloud server on any device that has a web browser. This includes the IPad/iPhone and Android devices.

### A. Overview of the Raspberry Pi

The proposed system, uses Raspberry Pi Model B+ single board computer and offers these key features:

- Broadcom BCM2835 SoC processor with 700MHz ARM1176JZF-S core

- 512MB RAM
- Videocore 4 GPU supports up to 1920x1200 resolution
- 5Mpix Camera module capable of full HD video @ 30fps
- MicroSD card slot, 10/100Mbps Ethernet port, 4 x USB 2.0 ports, HDMI, audio/video jack, GPIO header, microUSB power port, DSI and CSI ports
- Dual step-down (buck) power supply for 3.3V and 1.8V

During initial setup Raspberry Pi was configured as a miniature desktop with USB configured camera module and an external monitor or mobile for viewing the captured video. The Raspberry Pi runs Raspbian OS and is programmed using GNU Octave and Python, which is an open source. The Dynamic host Protocol is obtained for Raspberry Pi, addressed IP is fetched. After obtaining this the system can be configured and controlled remotely.

### *B. Connecting the Camera*

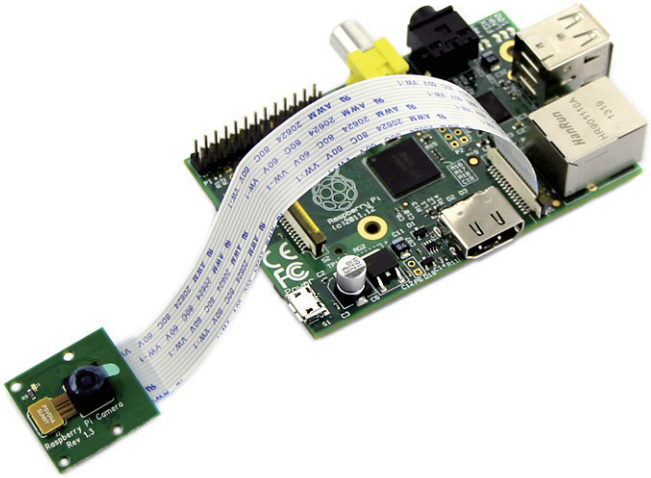


Fig. 2: Connection of Camera Module on Raspberry Pi board

As shown in Figure 2, a 5MP camera module that capable of 1080p video and still image, and it can connect to Raspberry Pi directly with CSI (Camera Serial Interface). And then boot the latest version of Raspbian and we are good to go with the camera. Its a fixed focus 5MP sensor capable of 2592x1944 stills, but also 1080p30, 720p60 and 640x480p60/90.

Cost of the whole system is around $75. Includes the camera module costs, Raspberry Pi board and wifi adapter. So the system cost is less, compared to other alternatives.

## III. Motion Detection Algorithm

Motion detection works on the basis of frame differencing - meaning comparing how pixels (usually blobs) change location after each frame. The method looks for a object change in the image:

The problem with these motion detection methods is that neither detects slow moving objects, determined by the sensitivity of the threshold. But if the threshold is too sensitive, it will detect things like shadows and changes in sunlight. The Motion Detection algorithm also cant handle a rotating object - an object that moves.

**Algorithm 1** Motion Detection algorithm

1: **procedure** MotionDetection
2: *calculate the average of a selected color in frame 1*
3: *wait X seconds*
4: *calculate the average of a selected color in frame 2*
5: **if** $abs(avgFrame1 - avgFrame2) > threshold$ **then**
6: *motion detected*

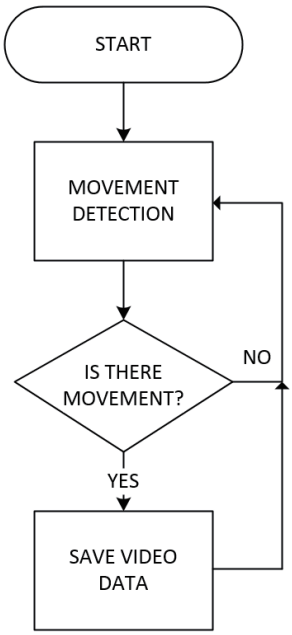


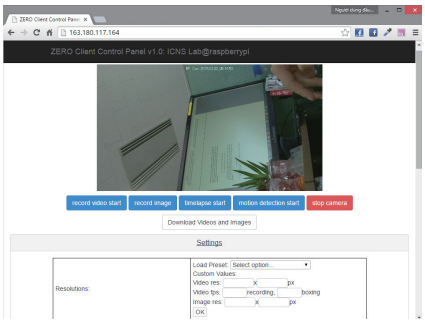

Fig. 3: Motion Detection Flowchart

Motion Detection flowchart is shown in Figure 3. If there is no motion detected, the program will not save the videos data. Otherwise, if motion is detected, the current frame of detected motion will be processed by Motion Detection algorithm. And then the system will be record full-hd videos, save them on the sd-card packed into MP4 container while the live-preview continues.

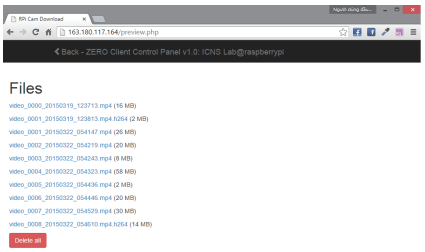
## IV. Experiments & Results

Figure 4 presents the screenshots of the system. The control panel with control options to save stills and full-res MP4 file, capture a picture, timelapse, change the resolution of videos (Figure 4a). When movement occurs, the system will analyze incoming images and store the most importance items. Go to the "Download Videos and Images" link, we can view a preview of the JPEG images and videos play smoothly (Figure 4b, 4c). We also can watch videos on mobile devices, it's got a smooth set of controls, reliable performance, and a clear picture with no blips, glitches (Figure 4d).

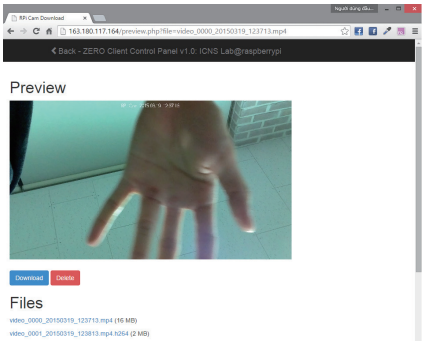
The latency remains fairly low at about 0.4 seconds for motion detection based on comparing an on screen timer and the video image of this system. While remotely viewing the 640x360 MJPEG image, the Raspberry Pi reports 67% CPU. But it's not overclocked.



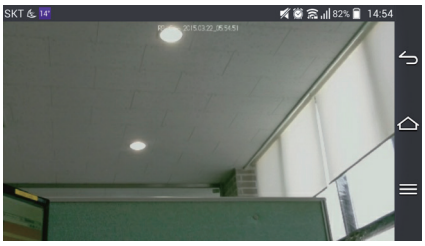

(a)




(b)




(c)



(d)

Fig. 4: System screenshots: (a) monitoring web interface; (b) videos list after recorded; (c) preview videos real-time; (d) interface on mobile devices

TABLE I: Comparison of data storage between with and without motion detection

| | Storage Space (GB/month) |
|---|---|
| Non-Motion Detection | 236 |
| Motion Detection | 740 |

Table I presents a comparison of the storage space between with and without motion detection. Storage space when using motion detection is 236 GB/month, while the we take 740 GB/month if not using motion detection. The motion detection algorithm would use help to minimize the recorded data storing capacity. It means that we can save money for the storage fee.

## V. Conclusion & Future Work

An approach for the surveillance monitoring system and the Motion Detection algorithm to decrease storage usage based on Raspberry Pi single board computer was proposed in this paper. This is still far from being final version of the monitoring system and there is lot for us to improve.

In future we plan to improve the Motion Detection algorithm. Because the algorithm depends on threshold value, it's mean the performance of the algorithm can be enhanced by considering certain conditions. If we have a good solutions to get a good threshold value and then the algorithm can detect moving objects precisely, including slow moving or tiny objects.

## VI. Acknowledgments

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2010-0020725). This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-(H8501-15-1015) supervised by the IITP(Institute for Information & communications Technology Promotion)). Eui-Nam Huh is the corresponding author.

## References

[1] Yong-ik Yoon, Jee-ae Chun, *Tracking System for mobile user Based on CCTV*. Information Networking (ICOIN), 2014 International Conference on, Phuket, 10-12 Feb. 2014, pp. 374-378.

[2] Viren Pereira, Vandyk Amsdem Fernandes, Junieta Sequeira, *Low Cost Object Sorting Robotic Arm using Raspberry Pi*. Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), 2014 IEEE, Trivandrum, 26-27 Sept. 2014, pp. 1-6.

[3] Yimamuaishan.Abudoulikemu, Yuanming Huang, Changqing, *A Scalable Intelligent Service Model for Video Surveillance System Based on RTCP* . Signal Processing Systems (ICSPS), 2010 2nd International Conference on (Volume:3 ), Dalian, 5-7 July 2010, V3-346 - V3-349.

[4] C. Bahlmann, Y. Zhu, Y. Ramesh, M. Pellkofer, T. Koehle, *A system for traffic sign detection, tracking, and recognition using color, shape, and motion information*. IEEE Intelligent Vehicles Symposium, Proceedings,2005, pp. 255-260.

[5] Adrienne Heinrich, Dmitry Znamenskiy, Jelte Peter Vink, *Robust and Sensitive Video Motion Detection for Sleep Analysis*. Biomedical and Health Informatics, IEEE Journal of (Volume:18 , Issue: 3 ) , 2168-2194, 20 September 2013, pp. 790-798.