© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie

The International Journal of Integrated Engineering

A Design Methodology of an Embedded Motion-Detecting Video Surveillance System

Muralindran Mariappan[1*], Lim Kean Thong[1], Karthigayan Muthukaruppan[2]

[1]Faculty of Engineering,
University Malaysia Sabah, Kota Kinabalu, Sabah, MALAYSIA

[2]PG INTSYS SDN BHD
Petaling Jaya, Selangor, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2020.12.02.007
Received 1 March 2019; Accepted 2 January 2020; Available online 28 February 2020

Abstract: Malaysia urbanization rate has been growing fast due to rapid development and modernization (Economy Planning Unit, 2015) (World Bank, 2015). With the fast growing urban population, the property crimes associated also rises at an alarming rate (United Nations Human Settlements Program, 2007) (Mohit Mohammad Abdul, Elsawahli H. Mohamed Hassan, 2015) (Ministry of Home Affairs, 2015). Video surveillance system is one of the trusted and efficient security systems to protect people and property against criminal (Varij Ken, 2015). Three problems and issues regarding current video surveillance system are functionality, flexibility and efficiency. The aim of this project is to design and develop an embedded motion detecting video surveillance system and to produce a functioning prototype as the final result to solve these problems. In order to achieve the aim of this project, three objectives are set to be attained. These objectives are to design and develop a motion detection algorithm based on OpenCV library functions using camera as sensor, to process and execute the algorithm using an embedded micro-computer and to compare its processing performance with a PC and to enable wireless user alert using LAN and internet connection. The first objective is achieved as the motion detection algorithm designed and developed based on background subtraction method using camera sensor and openCV library function is proven to functioning well in detecting for motion changes. The algorithm performs successfully both in BeagleBone Black module and PC and is able to deliver outputs required for embedded motion detection video surveillance system. The second objective is achieved as BeagleBone Black (BBB) module, a micro-computer embedded system is used as main processor for the embedded motion-detecting video surveillance system. The BBB module is able to process and execute the motion detection algorithm designed with image processing functions from OpenCV library. The third objective is achieved as embedded motion-detecting video surveillance system is equipped with wireless user alert function using local area network and internet connection. The system is able to send real-time alert to the user via email attached with the image captured and detected as a threat by the system

Keywords: Embedded Motion-Detecting, Video Surveillance System, Detection and Tracking System

1. Introduction

Functionality is the first criterion to be fulfilled by any product design. Property crime prevention saves valuable lives, money and time. Video surveillance system commonly uses closed-circuit television (CCTV) to record live scene of camera covered area in the house which helps user to monitor their house and property. However, the CCTV functions only as a video recorder by storing the images or videos into hard drive. It will be helpful in assisting police and investigator to solve crime by providing recorded evidence of offenders' identification and crime acts (Rudgard Olivia,

*Corresponding author: murali@ums.edu.my

2015). However, this poses a problem where property crime prevention is not succeeded as harms and damages have already been done. Video surveillance system should operate not only in video recording but also detecting and identifying threats such as house break-ins, burglary and robbery in the area covered.

Mobility and flexibility are critical design factors. Current market available surveillance system using internet protocol (IP) camera do provide solutions to the problem mentioned above where user will be alerted with alarm if any motion detected or person breaks into defined area covered by the sensor. However, these systems often require a personal computer to initiate, operate and manage by using software or program provided by the manufacturer (Wollerton Megan, 2015). This raises flexibility issue where a general purpose PC has to be dedicated for the surveillance system to function and operate. User would need to purchase a PC solely to support the system and could not move it anywhere else which is highly inconvenient and inefficient. Video surveillance system should be able to operate and function independently. This solution is feasible by using a lower cost embedded micro-computer to replace the PC.

Efficiency is about minimizing the effort to achieve desired result. Market available video surveillance systems are mostly based on wired communication with built-in house alarm (Vinay Sagar K. N., Kusuma S. M, 2015). Gesture recognition using image processing and IOT Linux embedded system where achieved lower than 5% (FREIRE, Davi Soares et al, 2018). Once a threat is identified by the surveillance system, it will trigger and sound the alarm to alert the owner or user inside the house. However, this raises a problem where property crimes are most probably to take place when the house owner is away from home and therefore the surveillance system would fail to alert the user immediately. Crime prevention is unsuccessful in this circumstance. Therefore, video surveillance system should be able to communicate with users wirelessly in providing near real-time alert and information regardless of their location and distance from home. Few challenges are faced in this project like object recognition are considered.

2. Methodology

Embedded motion-detecting video surveillance system design concept is illustrated in Figure 1

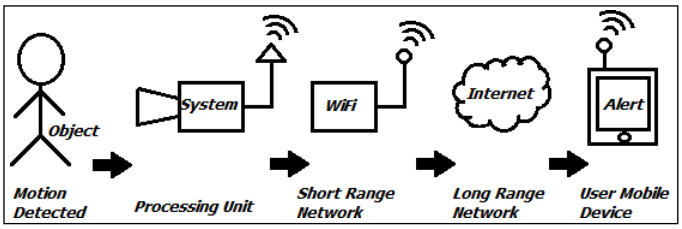

Fig. 1 - Embedded motion-detecting video surveillance system's methodology block diagram

The system takes object motion as input and process. If threat is detected, the alert information is transmitted to the user's mobile device using WiFi as short range network and Internet as long range network. The system can be broken down into three parts with each part is specifically derived to achieve all three project's objectives respectively. The three parts are: motion detection, processing unit and wireless alert. Figure 2 shows the embedded motion-detecting video surveillance system's methodology block diagram

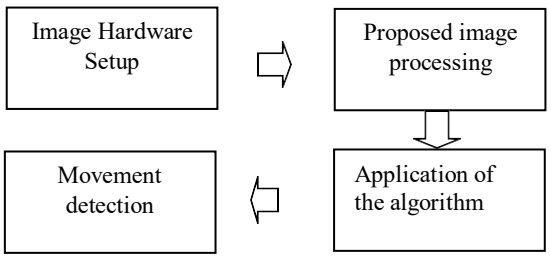

Fig. 2 - Embedded motion-detecting video surveillance system's methodology block diagram

The system starts with execution of program. At any point of the program, user will be able to stop and end the program using keyboard input via interrupt function. First, the system will check for sensor status, which is a webcam. USB webcam is chosen to be the hardware sensor because it is low cost, widely available and user friendly. If the sensor is not detected or not functioning, a notification will be displayed on system console and the program will stop and end. If the sensor status returns true, the system will proceed to setting the frame size. The frame size is can be either set default by the camera or manually set by the user. For this prototype, the resolution is 320 pixels in width and 240 pixels in height recording in YUYV color image format. The default capturing speed is set to 30 fps and is adjustable.

After setting the frame size, the system will update the date and time using network time protocol. The Network Time Protocol (NTP) is used to synchronize the clocks of machine with public internet and private networks. The synchronization protocol determines the time offset of the server clock relative to the client clock (Carroll Lewis, 2012). NTP synchronizes all participating computers to within a few milliseconds of Coordinated Universal Time (UTC) (Mills David L, 2010). The time of a client relative to its server can be expressed in equation 3.12 where t is the current time, T is the time offset at the last measurement update t0, R is the frequency offset and D is the drift due to resonator ageing.

$$T(t) = T(t_0) - R(t - t_0) + \frac{1}{2} D\,(t - t_0)^2 \quad (1)$$

The system then proceeds to setting restricted region. Restricted region is the region where any motion detected will be defined as a threat. The restricted region set will be highlighted on display in the video. The system then proceeds to capturing images. In OpenCV library, videoCapture.open is used to open & check the capturing device for video capturing, videoCapture.read is used to grab, decode video frame. If no frames has been grabbed i.e. camera has been disconnected, or there are no more frames in video file, the methods return false and the functions return null pointer. Two frames will capture one after another.

The first process is gray-scale. Gray scale is an image processing technique such that the output image contains only intensity information regardless of its input color components. It is composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest (Johnson Stephen, 2006),. The video capture pixel format used by the sensor is in YUV format. YUV color encoding model encodes brightness information separately from color information. In YUV format, Y is the luminance or brightness component derived from linear RGB values, Y‟ is the luma component derived from non-linear RGB values, U and V are the chrominance or color components. Luma is derived from an RGB color by taking a weighted average of the red, green, and blue components as shown in formula (2). In OpenCV library, cvtColor and COLOR_BGR2GRAY function is used to converts an RGB color image to gray.

$$Y' = 0.299R + 0.587G + 0.114B \quad (2)$$

Both images are then compared for absolute difference. Absolute differencing is an arithmetic operation performing in two arrays which is two images in image processing wise. When performing an arithmetic operation on two images, the operation is done in pixel-wise, i.e. the first pixel of image A is subtracted from the first pixel of image B, the second pixel of A is subtracted from the second pixel of B, and so forth (Levin Golan, 2016). The function takes the brightness values from both pixels and subtract to each other in order to obtain difference result. In OpenCV library, absdiff function is used to calculate the per-element absolute difference between two arrays. Absolute difference between two arrays of monochromatic arrays with same size is given formula (3) as shown below where src1 is first input array, src2 is the second input array and dst is the output array.

$$dst\,(I) = saturate\,(|src1\,(I) - src2(I)|) \quad (3)$$

The resultant image will then be filtered using threshold to obtain black and white only binary image. A thresholding operation chooses some of the pixels as the foreground pixels that make up the objects of interest and the rest as background pixels (Shapiro L., Stockman G, 2000). The threshold operation replaces pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant. In OpenCV library, threshold function is used to apply threshold to an array. The function transforms a gray scale image to a binary image according to the formula (4) as shown below where src is the input array, T is the threshold calculated individually for each pixel, maxValue is a non-zero value assigned to the pixels for which the condition is satisfied and dst is the output array.

$$dst(x,y) = \begin{cases} maxValue, if\ src(x,y) > T\,(x,y) \\ 0, otherwise \end{cases} \quad (4)$$

The binary image is then smoothed to eliminate noise. Image is often composed of some underlying ideal structure and some random noise. Noise which varies randomly above and below a nominal brightness value for a region can be reduced by averaging a neighborhood of values. Equation (5) defines a smoothing filter that averages 25 pixel values in a 5x5 pixels neighborhood of the input image pixel in order to create a smoothed output image where Out is the output image array, In is the input image array, r is row and c is column (Shapiro L., Stockman G, 2000). Box filter is used to smoothing an image equally by weighting a rectangular neighborhood of pixels to connect dots and smaller lump white color regions into bigger white color blobs. In OpenCV library, blur function is used to smooth an image using the normalized box filter. The function smoothes an image using the kernel as shown in equation (6) where kszie is the value assigned for blur size.

$$Out[r,c] = (\sum_{i=-2}^{+2}\sum_{j=-2}^{+2} In[r+i,c+j])/25 \quad (5)$$

$$K = \frac{1}{ksize.width * ksize.height}\begin{bmatrix} 1 & \dots & 1 \\ \dots & \dots & \dots \\ 1 & \dots & 1 \end{bmatrix} \quad (6)$$

The processed image is then searched for contours. Contour is the result of border following which derives a sequence of coordinates or the chain codes from the border between a connected component of 1-pixels (1-component) and a connect component of 0-pixels (background or hole) as defined in Figure 3. Border following is able to discriminate outer borders and hole borders by putting unique mark on each border (Suzuki S, Abe K, 1985). In OpenCV library, findContours function is used to detect and label the blobs. It retrieves only the extreme outer contours and compresses horizontal, vertical, and diagonal segments and leaves only their end points.

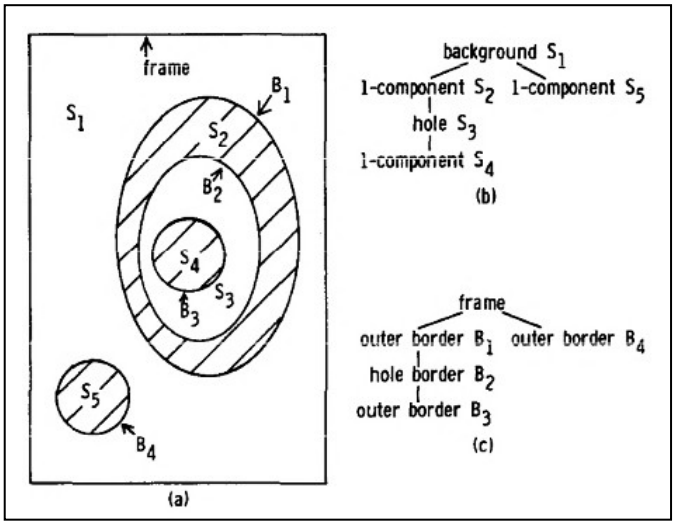

Fig. 3 - Suzuki's border following components, holes and respective borders definition

Any contour detected will be defined as object detected. Since only one object shall be focused in one single frame, the object is then determined by the biggest vector size of all the contours detected. The object is then locked and tracked for its position. In OpenCV library, boundingRect function is used to calculate the up-right bounding rectangle of the largest object's point set. This function is used to calculate the centroid of the object detected by using the equation of (7) where objrect is the vector point of bounding rectangle starting from (0,0).

$$(x,y)_{centroid} = objrect.(x,y) + objrect.\frac{(width,height)}{2} \quad (7)$$

By calculating the centroid of the object, the position is then determined and locked in order to trace the object whether it enters the restricted region defined by the user. The restricted region is a set of rectangle (x, y) location in the image frame using if conditioning command as shown below where x1 is the starting and x2 is the ending number of columns in frame's width while y1 is the starting and y2 is the ending number of rows in frame's height.

```
if (x_centroid >= x1 && x_centroid <= x2)
        if (y_centroid >= y1 && y_centroid <= y2)
                threatdetected = true;
```

The image frame reference is as shown in Figure 4 where column and row (x, y) starts from upper left corner for a 320 x 240 pixels image.

(0,0)	(1,0)	...	(319,0)	(320,0)
(0,1)	(1,1)	...	(319,1)	(320,1)
⋮	⋮	⋮	⋮	⋮
(0,239)	(1,239)	...	(319,239)	(320,239)
(0,240)	(1,240)	...	(319,240)	(320,240)

Fig. 4 - Image reference point for a 320 x 240 pixels of image with each box representing one pixel

The position of the largest object will be checked for its position. If it is inside the restricted region, threat detected will return true and the system will send an email alert containing the image frame to the user's email address. The alert approach chosen in this project is e-mail using Gmail mail server and SSMTP program as the deliverer. User of embedded motion-detecting video surveillance system is able to receive alert upon threat detection using their mobile phone or any internet connected machine. An e-mail account with address limkeanthongdevice1@gmail.com is created and assigned for the system as sender and the user or the recipient e-mail address is required as well. The content of the e-mail is an image of threat detected frame recorded by the system. In order to avoid spamming, an interval of 30 seconds is set for every alert e-mail to be sent to the user. In order to achieve global wireless connection, internet is the best long-range wireless medium considering the cost and availability for implementation. The BBB module does not equipped LTE module for wireless communication of high-speed data transmission. Therefore, a local area network LAN connecting to home fixed-line telecommunication router with internet connection is needed. The WiFi adapter used in this project is TP-Link TL-WN722N high gain wireless adapter to enable BBB module to connect with router.

The restricted region will be highlighted on display upon detection of threats. The image frame will be recorded into video if object is detected regardless of threat. In OpenCV library, imwrite function saves the image to the specified file. The image format is chosen based on the filename extension, e.g. .jpg for JPEG files, .png for Portable Network Graphics. Only 8-bit single-channel or 3-channel with BGR channel order images can be saved using this function. Meanwhile, the function videoWriter.write writes the specified image to video file. It must have the same size as has been specified when opening the video writer. The video writer requires information such as fcc, the four character code of a video standard format e.g. DIV3, MPG4 and FMP4, fps, the video play back speed calculated in frame per second, framesize, the image size to be written and videocolor for color option.

The algorithm then repeats itself starting from capturing images from the camera sensor. The processing unit used in this project is Beagle Bone Black (BBB) module by BeagleBoard.org. BBB is an embedded micro-computer module powered by ARM Cortex-A8 32-Bit RISC processor with 1 GHz of processing speed. The USB webcam driver used in this project is uvcvideo. The USB Video Class, defines video streaming functionality on the Universal Serial Bus. UVC compliant peripherals only need a generic driver that supports video streaming for both video input and output (Linux Kernel, 2016). The format of video capture is set to be 320 in width and 240 in height for image frame size. This limitation is due to slow data transfer speed on USB 1.1 bus. Equation (8) shows the data transfer speed in a bus where width and height is the frame size, channel is the color component and FPS is frame per second.

$$Data\ Transfer\ Speed = Width\ x\ Height\ x\ Channels\ x\ FPS \quad (8)$$

For an uncompressed YUV image at 320 x 240 pixels frame size, 3 channels for 'RBG' color components and FPS set to 30, the data transfer speed is as shown in equation 9.

$$Data\ Transfer\ Speed = 6.91\ Mbits/s \quad (9)$$

The bus speed for USB 1.1 ranges from low speed at 1.5 Mbits per second to full speed at 12 Mbits per second maximum (Pan Hui, Polishukl Paul, 1998). Therefore, the frame size of image captured by USB webcam must set to 320 x 240 pixels frame size in order to allow BBB module to receive the data via the driver installed.

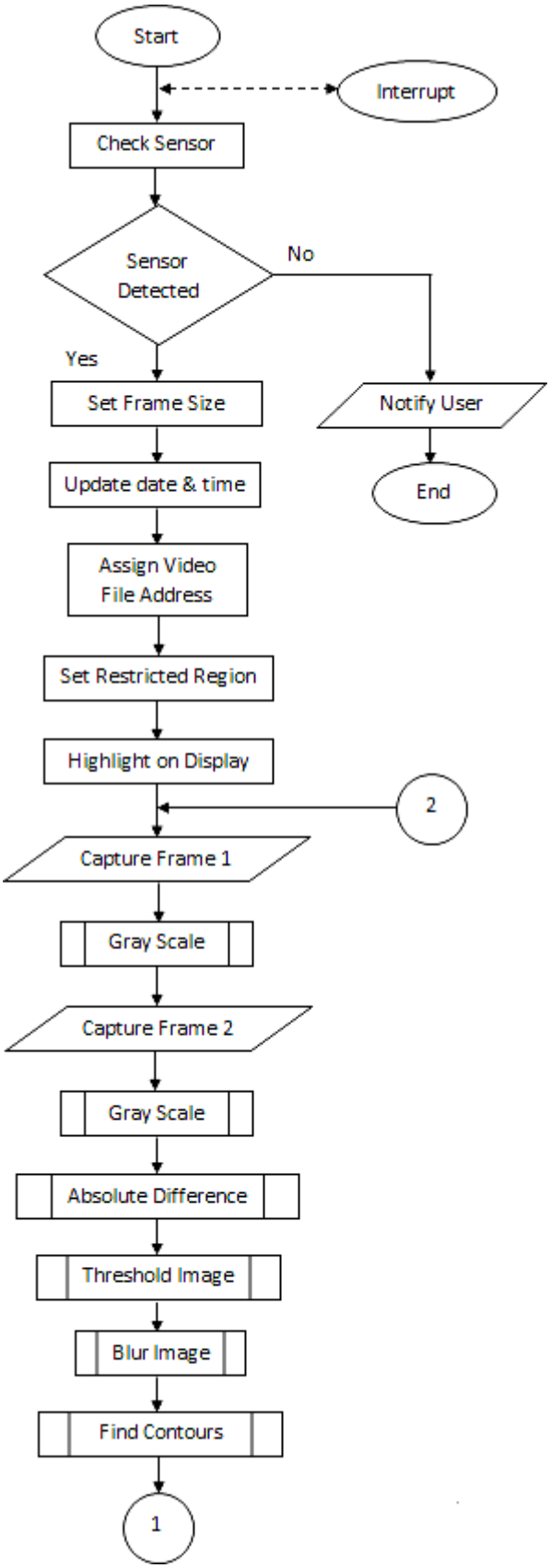

Fig. 5 - The process flow of the system

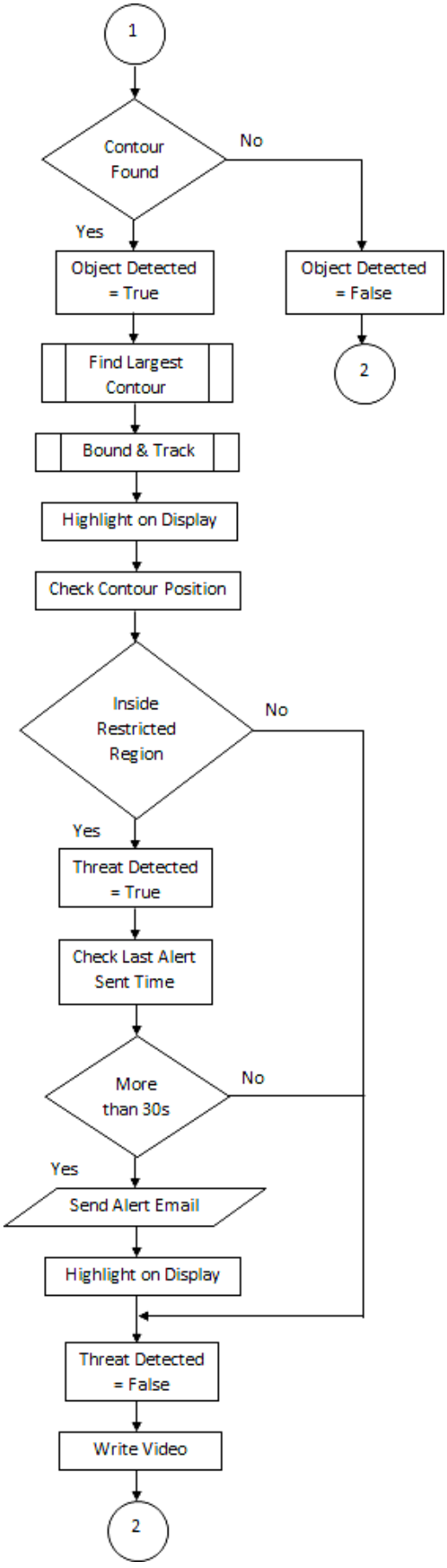

Fig. 5 - The process flow of the system

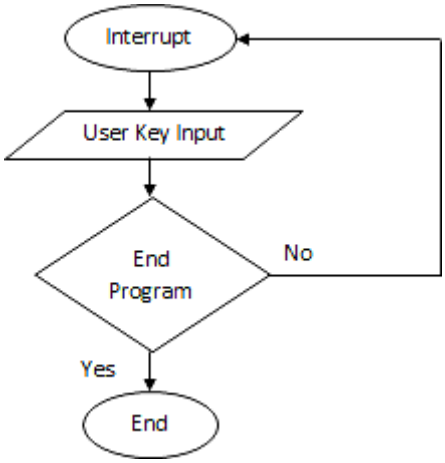

Fig. 5 - The process flow of the system

3 Results

3.1 Motion detection

The motion detection algorithm is tested to obtain the processed image results. The tested video is carphone test video by The Stanford Center for Image Systems Engineering website (Stanford University, 2016). Two images are captured using video capturing function. Figure 6 shows the captured original image and Figure 7 shows the captured moved image.

Images are gray scaled to eliminate color components. Figure 8 shows the gray scaled original image and Figure 9 shows the gray scaled moved image. As can be observed in the images, the colors components are all removed and left gray scaled into brightness components only. Images' pixel brightness is subtracted to obtain absolute difference image. Figure 10 shows absolute difference image. The subtraction results a value ranging from 0 to 255 of 8-bits data. The absolute black region shows 0 and absolute white region shows 255. The gray region is the values in between this region. The difference in brightness values indicates changing in motion. As can be observed in the figure, the edges of head and body of the man are brighter compared to static background due to movement and motion.

Fig. 6 - Captured original image

Fig. 7 - Captured moved image

Fig. 8 - Captured moved image

Fig. 9 - Captured moved image

Fig. 10 - Absolute difference image

The image is threshold to obtain binary image. Figure 11 shows threshold image. Binary image is in the normalized digital format in either 255 or 0 value. Digital format is acquired since analog format is not required in detecting motion in this algorithm. As can be observed in the figure, the gray values are threshold and removed, either normalized to 255 or 0, based on the threshold value set by user.

Fig. 11 - Threshold image

The image is blurred to connect all the smaller white spots detected as noise. Figure 12 shows smoothed image. As can be observed in the figure, the smoothed image has gray value in the regions between absolute black and absolute white.

Fig. 12 - Smoothed image

The image is threshold to obtain binary image. Figure 13 shows second threshold image. Second threshold is applied to remove the gray regions and therefore eliminating smaller and discrete noise by merging it into bigger white region as can be observed in the figure.

Fig. 13 - Second threshold image

The image is detected for contours and the edges are labeled. The vector points are used to calculate the objects sizes and centroid.. The largest object detected is bounded and the calculated centroid in x and y position is displayed in console. The centroid is highlighted with a green marking. Figure 14 shows the object's centroid x and y position and Figure 15 shows the highlight of detected in motion object. As can be observed in the figure, the head of the man is detected as the largest moving objected detected and its centroid position is 81 in x and 16 in y.

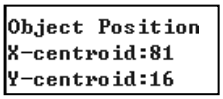

Fig. 14 - The detected object's position

Fig. 15 - Highlight of detected object

3.2 Image Quality Test

In order to measure the image quality processed by the BBB, the processed images are compared to images processed by a PC powered with Intel Core-i5 2450M using MSU Video Quality Measurement Tool software by measuring the Peak Signal Noise Ratio (PSNR) value. Three functions of image processing are tested: writing video, absolute differencing and thresholding. The Carphone test video is used as the sample for this measurement. Table 1 summarizes the average PSNR value measured for three functions. As can be observed from the table, the average PSNR value obtained in writing video is 42.35587 dB which is in the range of typical 30 dB to 50 dB PSNR value for a lossy image with 8-bit depth. In absolute differencing functions and thresholding functions, the PSNR values obtained are 23.82315 dB and 22.66375 dB respectively, which is slightly lower than typical range. In overall, the BBB underperforms in image processing comparing to PC, however, is able to process and produce image with adequate quality for embedded motion-detecting video surveillance system.

Table 1 - Average PSNR value measured three functions

Functions	Average PSNR (dB)
Writing Video	42.35587
Absolute Differencing	23.82315
Thresholding	22.66375

3.4 Prototype Functionality Test

The final result of this project is to produce a functioning prototype of embedded motion-detecting video surveillance system. Figure 16 shows the prototype of the system. As can be observed, the BBB module serves as processing unit connecting to wireless adapter, camera sensor and power supply. The prototype of embedded motion-detecting video surveillance system is tested in three scenarios for prototype system functionality. The first real life scenario location is at the corridor where a door is set as the restricted region. The second real life scenario is at the porch where the motorcycle parking space is set as the restricted region. The third scenario is inside a house where the entrance is set as the restricted region.

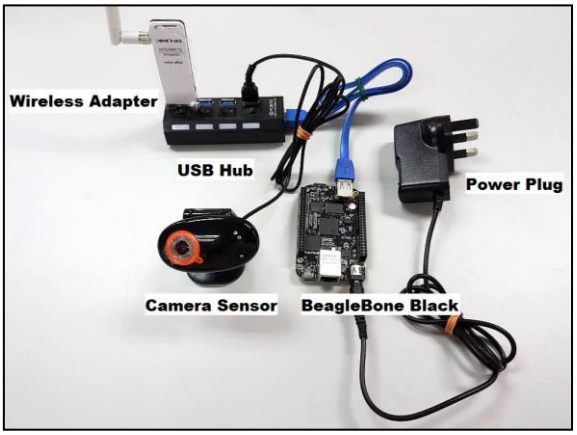

Fig. 16. - Prototype of embedded motion-detecting video surveillance system

Figure 17 shows the information status display by the system when running the system. The system display "System started" indicating the system has been initialized and started. The updated time and date is then displayed which is 8th of June 2016 at 1930 time. The address of the video recorded is at the location /media/B8A4-3F50 which is the SD card name. The system then continues running and display information such as "threat detected" and "alert sent to limkeanthongdevice1@gmail.com" if the condition is triggered and run by the system.

```
login as: root
Debian GNU/Linux 7

BeagleBoard.org BeagleBone Debian Image 2014-05-14

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
Last login: Wed Jun  8 19:24:27 2016 from 192.168.7.1
root@beaglebone:~# ./test9
SYSTEM: System started
SYSTEM: Date & time = 2016-06-08 @ 19.30
SYSTEM: Video created at location /media/B8A4-3F50/
SYSTEM: Threat Detected!
SYSTEM: Alert sent to limkeanthongdevice1@gmail.com
SYSTEM: Threat Detected!
SYSTEM: Threat Detected!
SYSTEM: Threat Detected!
SYSTEM: Threat Detected!
```

Fig. 17 - Information display by the system showing the status

Figure 18 shows the threat detected image sent to the user for real-life scenario test 1. In this image, a man is approaching a door which is the restricted region set by the user. This action is considered as a threat and this image is captured and sent to user's email. Figure 19 shows the threat detected image sent to the user for real-life scenario test 2 which sets the motor parking region is restricted area while Figure 20 shows the threat detected image sent to the user for real-life scenario test 3 which sets the door inside the house or room is restricted region.

Fig. 18 - Threat detected image for real-life scenario test

Fig. 19 - Threat detected image for real-life scenario test

Fig. 20 - Threat detected image for real-life scenario test

4. Discussion

4.1 Challenges & Solutions

One of the challenges for embedded motion-detecting video surveillance system project is the low image resolution. According to the manufacturing specification, the sensor used in this project, Sensonic Webcam VC7000, is able to capture high definition image up to 1280 x 720 pixels. High definition image is better for motion detection image processing in term of accuracy since more image pixels equals to more signal pick up. Object will then be easier to be identified for shapes and sizes. In order for this sensor to deliver high definition image, a Microsoft OS based software needed to be installed to compress the image to be delivered via USB 2.0 driver bus and reconstruct when the compressed image is delivered into the processing unit. However, the camera sensor driver software does not support Linux based operating system. The driver installed for this sensor is uvcvideo, which is a universal driver that does not support image compression. Thus, the processing unit does not support uncompressed high definition or high resolution image input directly via USB 1.1 driver for image processing.

The solution to solve low image resolution challenge is to process the image in more detail ways. The 320 x 240 pixels image is, however, sufficient to detect for large object such as human, which fulfill the requirement for motion detection. For smaller object, potentially such as dog, cat, leaves and etc., will be misinterpreted as human, the larger object. Thus, the image filtering is needed to eliminate error as much as possible. Image processing functions such as thresholding and smoothing are needed to filter out noises and focus on larger object to get more accurate result. Thresholding allows smaller noises to be filtered out while smoothing allows smaller and nearer white blobs to be merged to create a larger blob. Considering only the moving parts are detected as blobs, human normally will move several parts such as head, body, hands and legs together when doing activities such as walking. These smaller blobs can be be joined together to form a larger blobs using smoothing functions, therefore, creating a human shape figure. By adjusting the sensitivity and blur size, based on the scene or scenario required, large object or human can be detected under 320 x 240 pixels image, therefore solving the lower resolution problem.

The second challenge is for embedded motion-detecting video surveillance system project is the implementation of image processing in embedded system. Personal computer is preferred when doing image processing work because of its higher speed and higher performance processor as compared to embedded system. Processor with higher performance is able to process image more accurate and produce a higher quality image. This is proven in the result comparing the BeagleBone Black processed image quality and the PC processed image quality. The image processed by PC is averagely higher value in PSNR for both threshold and smoothed images as compared to BBB. Higher PSNR indicates higher image quality. In term of speed, the processing time required for PC is shorter since the processor speed for PC is 2.5 GHz as compared to 1.0 GHz processor speed for BBB. More time is required to complete a main loop function for motion detection algorithm thus delaying frame queue time for video recording in BBB module. In term of user interface experience, BBB module does not equipped with user interface devices such as monitor and keyboard like PC does. Lack of these facilities restricted and delayed the development of motion detection algorithm. Debugging process during programming could not be done in real time.

The solution to solve image quality challenge is to ensure the processing power of BBB's processor is fully allocated in processing the image. No other functions or interruption should be run parallel with the program to ensure the maximum performance delivered by the processor in processing images. The time delaying for frame queue is

solved by adjusting the frame per second speed of the video to be recorded. By estimating the time required to complete a main program loop, the fps of the video can be adjusted using try-and-error method. Current fps rate set for the embedded motion-detecting video surveillance system is eight frames per second. The lack of user interfaces devices of BBB module is solved using PuTTY and WinSCP software tools. PuTTY is a free open-source terminal emulator and serial console which allows PC to control the BBB system in cross platform environment. WinSCP is a free open-source software to securely transfer file between local and remote computer which allows PC to debug the image processed by BBB without having to transfer the image remotely using storage devices. This software tools certainly help shorten the development time required for BBB module.

The third challenge is internet connection. For embedded motion-detecting video surveillance system to send as near-real-time as possible information or alert to user, internet is one of the best several options for wireless information exchange medium since internet is well-established, widely available and low cost. The BeagleBone Black module does not equipped with LTE or other telecommunication enable module which allows BBB to transmit data wirelessly.

One of the methods to connect BBB to router is via LAN cable. The advantages are the cable connection is always stable with minimum loss of signal and the data transfer speed is high due to optical fiber medium used in cable transmission. The disadvantage of this method is limited mobility for the prototype since cable has to be always connected between the BBB module and the router. The second way is via wireless local area connection to fixed line home router. The advantage of this method is high physical mobility for the prototype to be installed in anywhere within the network covered area. The disadvantage of this method is loss in signal is high as due to wireless transmission. The third way is via LTE network. The advantage of this method is the prototype is independent of local area network. The LTE module allows the BBB module to get online almost everywhere within the carrier's network coverage which is far wider than local area network can provide. The disadvantage of this method is a carrier line has to be dedicated and the cost for data is expensive for mobile network. Considering all the options available, the wireless LAN is chosen because of its higher mobility compared to via cable LAN and more cost efficient compared to via LTE network. The overall project success is 90% where the (FREIRE, Davi Soares et al, 2018), success rate is 95%, one of the key reason for higher success rate is due to platform of Linux. There are few areas are looking into increasing the percentage of the project.

4.2 Future Improvements

Based on the findings above, a few suggestions are recommended for future improvements. The first suggestion is to acquire higher resolution image. Considering on how acquiring higher resolution image would help getting more accurate result as discussed, it is highly suggested to use image compression enable camera. The image taken by the camera would be compressed into smaller size before sending to processing unit via USB Bus. The compressed image will then have to be uncompressed before any further processing in the processing unit. This improvement will benefit the image processing performance by producing higher quality of processed image and eventually leads to a better and more accurate result.

The second suggestion is develop mobile application user interface. Current prototype relies on PC for cross platform development and user interfacing. It is suggested to eliminate the reliance on PC and introduce mobile application UI for better user experience. As mobile phone is becoming essential part of everyday living, user to be able to monitor and control the embedded motion-detecting video surveillance system will be an advantage since this remote control solution offer convenience for the user without the need of PC. The mobile application can be developed to offer live video streaming function and to able to control the sensor to physically turn around for monitoring if requested.

The last suggestion is to add in identification functionality. The developed motion detection algorithm is able to detect for object but not recognizing the object. Considering some objects may not be a threat as seen by the user, it is recommended to add in identification function to further identify whether the object is friendly or threatening inside the restricted region before sending alert to the user. Identification function can be developed by building a database collecting all friendly object data. Through learning and calibration, the system will be able to identify object and respond accordingly.

5. Conclusion

In this paper, a detection and tracking system and hardware for video surveillance has been designed and developed using embedded system. The image processing aspect has been used in the detection the human in the given frame and the coding has been carried out using opencv tool which is widely applied in the computer vision area. The image processing includes smoothened, noise filter and thresholding to get the binary images. The embedded tool has proven that the human detection is successful carried out and able to draw box around the human through out in the screen. In order to benchmark the quality of processed image of BeagleBone Black and PC are compared. BeagleBone Black's processed image has adequate quality. The algorithm has been proven by running multiple testing in the

BeagleBone Black module and PC. The developed prototype is applied in the real scenario where it is able to alert the mobile using internet. The system is able to send real-time alert to the user. The further project direction has been discussed for improvement. The overall project success is 90%.

Acknowledgment
This research was supported by the Faculty of Engineering Project Grant. We thank our colleagues from the Electrical & Electronic Engineering Program who provided insight and expertise that greatly assisted the research.

References

[1] Carroll Lewis. (2012) "Executive Summary: Computer Network Time Synchronization", United States: University of Delaware, <www.eecis.ud'el.edu/~mills/exec>
[2] Economy Planning Unit(2015), "Eleventh Malaysia Plan 2016-2020: Chapter 10 - Malaysia Beyond 2020". Putrajaya: Prime Minister Department, <rmk11.epu.gov.my/index.php/en/dokumen-rmke-11>.
[3] FREIRE, Davi Soares et al. (2018), Embedded Linux System for Digital Image Recognition using Internet of Things. Journal of Mechatronics Engineering, [S.l.], v. 1, n. 2, p. 2 - 11, oct.
[4] Johnson Stephen. (2006), "Stephen Johnson on Digital Photography. United States: O'Reily Media, 122.
[5] Levin Golan. (2016) "Image Processing and Computer Vision". Open Frameworks, <openframeworks.cc>.
[6] Linux Kernel (2016), "The Linux Kernel Archives: Kernel documentation", The Linux Foundation, <www.kernel.org/doc/documentation/>.
[7] Mills David L. (2010), Computer Network Time Synchronization: The Network Time Protocol", United States: Taylor & Francis, 12.
[8] Ministry of Home Affairs, (2015) "Statistic of Violence Crime and Property Crime in Year 2011 to Year 2015 According to States", Putrajaya: Ministry of Home Affairs,
[9] Mohit Mohammad Abdul, Elsawahli H. Mohamed Hassan (2015). "Crime and Housing in Malaysia: Case Study of Taman Melati Terrace Housing in Kuala Lumpur", Asian Journal of Environment Behaviour Studies. Volume: 1, Number: 3.
[10] Pan Hui, Polishukl Paul. (1998), "1394 Monthly Newsletter. Information Gatekeepers", 7-9
[11] Rudgard Olivia. (2015) "Should You Install CCTV Outside Your Home", The Telegraph, <www.telegraph.co.uk/news/uknews/crime/11622935/Should-you-install-CCTV-outside-your-home>.
[12] Shapiro L., Stockman G. (2000) "Computer Vision", United States: Pearson, 275.
[13] Suzuki S, Abe K. (1985), "Topological Structural Analysis of Digitized Binary Images by Border Following". Computer Vision Graphics and Image Processing 30, 32-46.
[14] Stanford University. (2016). Test Images and Videos. United States: The Stanford Center for Image Systems Engineering,
[15] United Nations Human Settlements Program. (2007). "Enhancing Urban Safety and Security: Global Report on Human Settlements 2007". Kenya: United Nations, 2007, 68. <www.worldbank.org/en/news/feature/2015/01/26/malaysia-among-most-urbanized-countries-in-east-asia>.
[16] Varij Ken. (2015) "Malaysia Electronic Security Equipment Industry Outlook to 2018 - Increasing Demand for CCTV Cameras to Spur Growth. India", Ken Research Information Department, <kenresearchreport.wordpress.com>.
[17] Vinay Sagar K. N., Kusuma S. M. (2015) "Home Automation Using Internet of Things", International Research Journal of Engineering and Technology (IRJET). Volume: 2, Issue: 3, <scien.stanford.edu/index.php/test-images-and-videos>.
[18] Wollerton Megan,. (2015) "Delve Into DIY Security With These 35 Connected Cameras". CNET. www.cnet.com/news/security-camera-roundup.
[19] World Bank(2015), "East Asia's Changing Urban Landscape: Measuring a Decade of Spatial Growth". Washington DC: World Bank.