LabVIEW-based Motion Activated Security Camera

(MAS Cam)

Francisco I. Alano[1], Angelo S. Anzures[2], Jason G. Ondevilla[3], Mark Angelo C. Purio[4]
Electronics Engineering Department, Adamson University, Manila, Philippines
franciscoalano26@gmail.com[1], laninanjo@yahoo.com[2], jason_gabriel_45@yahoo.com[3], mpurio@yahoo.com[4]

Abstract— **This security system provides a cost-efficient alternative solution to intelligent security systems by using a standard webcam, personal computer, and a PIC microcontroller. This system uses a program developed in NI-LabVIEW which uses image processing. The said program only records the camera input once motion is detected, which reduces the duration of the video footage, disk space needed, as well as the time taken to review the recording. Furthermore, the camera continuously pans the area by using a PIC microcontroller and a servo motor. This feature allows the camera to cover areas outside its regular field-of-view.**

***Keywords*— motion detection, surveillance, security system, image processing, intelligent security system**

I. Introduction

The use of CCTV cameras is a widely implemented solution for preventing and solving crimes. The ability to identify perpetrators as their faces are recognized whilst reviewing the footage of crimes as they happen, or being able to prevent the whole thing as the mere presence of these systems make possible law breakers hesitate, or think twice about committing said crime. This has made CCTV footages a vital part of the investigative process. [1]

The growing demand for surveillance systems has led to major advances in CCTV systems. The conventional method of setting up a camera on a fixed location was no longer feasible for this demand. The need for intelligent visual surveillance systems and the importance for further improvements on that field has become one of the most important areas of research and development. For these systems to be fully implemented, it must meet real-time constraints such as time limits. [2] Therefore, these systems must focus on robust, real-time computer algorithms that can automatically adapt to changes in the environment like lighting, scene geometry, and scene activity. [3]

Intelligent surveillance systems may use a separate image processing software on a personal computer which can be used to communicate with a camera which is connected via wired or wireless media. The signals coming from the camera are translated into digital signals then sent to the host computer. [4] Image processing software can be used for communicating to external hardware such as a microprocessor unit, which controls the movement of the camera. [5]

The key factor in developing an intelligent security system is image processing. Image processing is a form of signal processing which is done on images which are usually digital. There are many various ways on how people use image processing, from simple photo editors to medical and security applications. Image processing is very common due to the advances in computer technology that the use of the term "to Photoshop" is very common that it proves how famous image processing is among the masses. [6] In image processing, an image is defined as a matrix that has image rows and image columns. Combining a row value and a column value is called a pixel. With this form of analysis, digital image processing methods can be used. [7]

There are many applications for image processing in intelligent surveillance systems. It can be used to indicate events such as video loss, fire alarms, and break-ins. These events can be identified by a set of pre-recorded IDs on the program. [8] This can also be used for combining multiple inputs from different cameras to produce a single large image without distortions. [9] One of the most common implementation of image processing in security systems is the method of image subtraction, where there are two consecutive frames in a video input are compared to track a moving object, which is presented in this paper.

II. Overview of the Design Project

The main focus of the design project is to develop a CCTV system which only records when it detects motion. This aims to reduce the duration of video files for easy reviewing, and minimize the stored files to save hard disk space.

The project consists of three major components: a camera, a personal computer, and a camera turntable. The camera will act as the input component for the program. The personal computer will act as the main component where all algorithms are processed, and as a data storage. When the data is processed, the computer then sends the corresponding data to the turntable. This turntable consists of a microprocessor that receives commands from the computer and a motor which controls the camera angle.

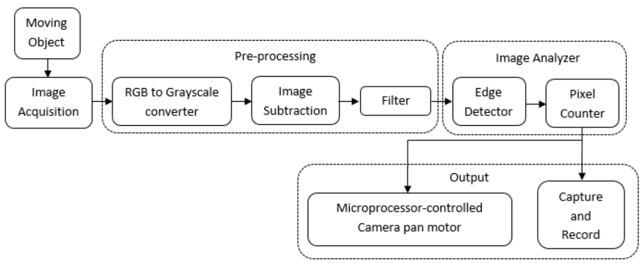

Fig. 1. Block Diagram of the Motion Activated Security Camera

Fig.1 shows the operation of the device and the program.

The USB camera and motion sensor detects the moving object and it will be processed by the LabVIEW program. The program commands depend on the condition: it will record footage to the PC and/or change the angle of pan motor.

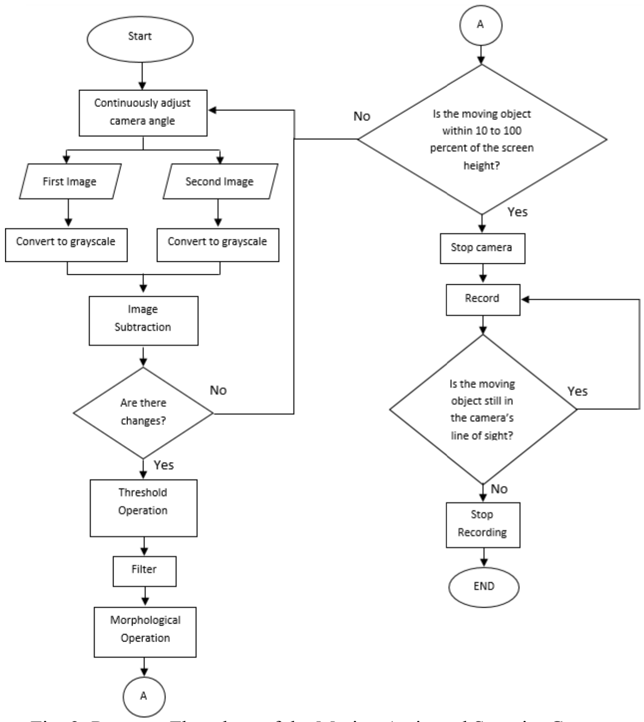

Fig. 2. Program Flowchart of the Motion Activated Security Camera

Fig. 2 shows the detailed process and operations of the program. The video sequence frames are converted to grayscale then filter to reduce noise of frames. It subtracts the current and previous frames. If there is a changes in the frame, it will simply further processing then extract information from video sequence and stores in the PC as a video file.

A. Program Implementation

The following images show the step by step procedure of the project, from image acquisition to filtering.

Image Acquisition

This step takes the image directly from the camera and copied to LabVIEW using the IMAQ driver. The color bits depend entirely on the camera. The recorded video will also be taken directly from here.

Fig. 3. Image Acquisition

RGB to Grayscale Conversion

The image is converted from colored (16 or 32 bit) to grayscale (8 bit). This is necessary for easy processing for the image subtraction. Fig. 4 shows the original image converted to grayscale.

Fig. 4. Grayscale image

Image subtraction and Threshold Operation

Two consecutive frames that are both in grayscale are compared by using image subtraction, and the result is considered the moving object. The sensitivity of the threshold operation is user controlled. For this image, the range of threshold is 5 to 50. The background is eliminated and all the pixels that changed are outlined in red, as shown in Fig. 5.

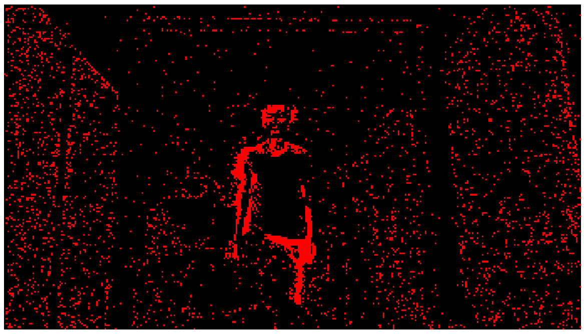

Fig. 5. Threshold Operation

Filtering

The subtracted image shown in the previous figure shows some elements from the background which may give a false positive, and cause problems like recording at the wrong time and the camera facing the wrong direction. These background noises are removed by filtering. The process of filtering takes three steps. The last step is to connect the separated parts object and consider it as a single moving object. This phase is done by using the Fill Hole function, which connects these parts into a single, large image. The resulting image is shown in Fig. 6.

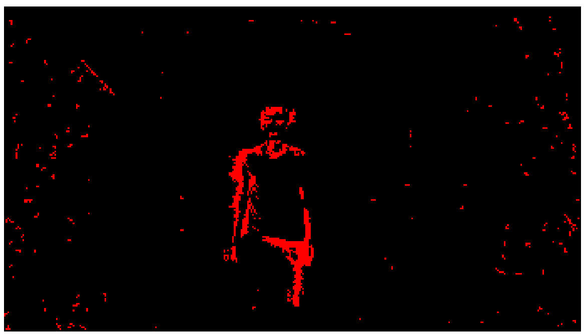

Fig. 6. Fill Hole function applied to image

'978-1-4799-8641-5/15/$31.00 ©2015 IEEE'

Fig. 6 shows the processed image when the Fill Hole is applied. The moving object is the only large object left on the screen, with minimal background noise.

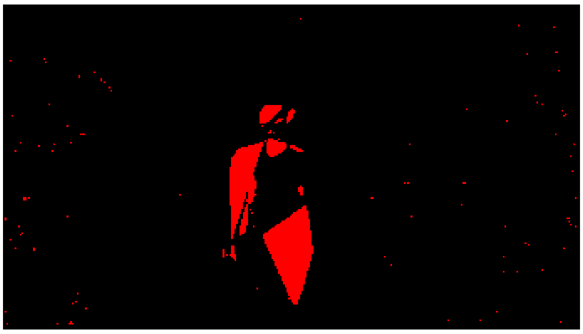

Fig. 7. Image with rejected border

The second part of filtering involves the removal of extra pixels that touch moving object. This involves the use of the Reject Border function in LabVIEW, which removes these extra pixels and produce a clear image. The result is shown in the Fig. 7 above.

The last step is to eliminate the remaining background noise. This phase is done by using the Remove Particle function, which removes the extra pixels in the background. The resulting image is shown on Fig. 8, and is ready for the next step of the operation.

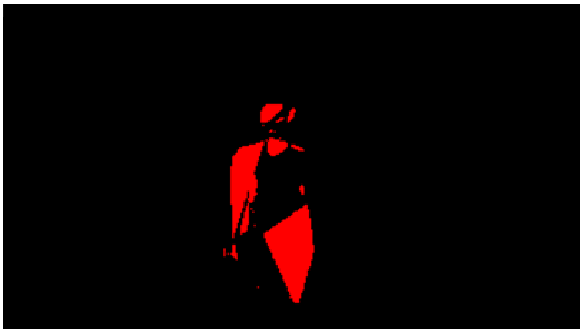

Fig. 8. Filtered image

Once the moving object matches the required parameters to identify that the moving object is an intruder, the program begins to record and save the video on the PC. To do this, a morphological operation is used for edge detection. This determines the size of the moving object, and the program compares the identified size to that of the set parameter. When these conditions are achieved, the recording will begin.

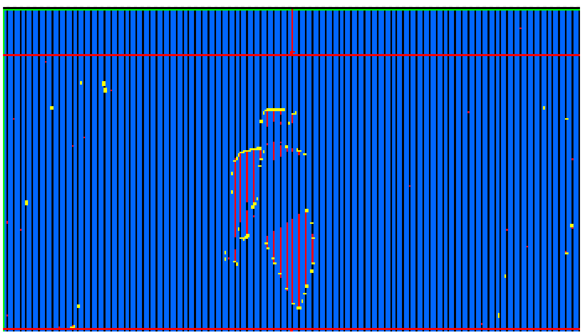

Fig. 9. Morphological Operation

Another feature of this project is that the camera continuously pans slowly until it detects a moving object. The camera stops when at its current position when a moving object passes through the camera's line of sight. When the camera does not detect any motion, the recording stops and the camera begins scanning the area again.

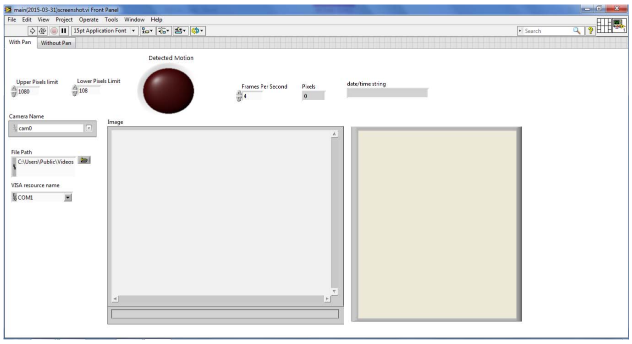

Fig. 10. LabVIEW Interface

Fig. 10 shows the user interface of the program. The large frame on the left shows the image currently being captured by the camera, and the one on the right shows the image with the morphological operation applied. The frame rate is set by the user which should match the frame rate of the camera used. The lower and upper pixel limits are also set on the camera being used. The upper pixel limit is the maximum pixel height of the camera. Since the camera uses a resolution of 1920 x 1080, the upper pixel limit is set to 1080 pixels. The lower limit is set in accordance to the CCTV Operation Requirements Manual [10], which states that detection will be triggered when a moving object occupies 10 percent of the screen height. The camera used has a screen height of 1080 pixels, therefore the lower limit is set to 108 pixels. An indicator LED will be lit when motion is detected. The video filenames use the date and time set on the computer for convenience.

B. Computer Hardware Specifications

- Processor: Pentium III/Celeron 866 MHz or equivalent
- RAM: 256 MB
- Operating systems supported: Windows 8/7/Vista (32-bit or 64-bit)

  Windows XP SP3 (32-bit)
- Disk space: 50MB for program

  At least 300GB for a 24-hour period total CCTV recording

These specifications are based on the system requirements of LabVIEW 2012, which was used in developing the program. The processor and RAM requirements were also based on testing the program on the lowest possible hardware specifications that does not affect the performance and reliability of the program. The minimum disk space for storing the video files was obtained through the bitrate of the video recording, which is 3.5 MBps.

III. Results and Discussion

The following data and results were obtained using a distance of 1150 cm, which is the maximum diagonal distance of the room were the testing was performed, and the distance where the moving person was at during testing. This distance was set to ensure that the person will be detected as soon as he/she enters the room. At this distance, the height of the captured image is 250 cm, which captures the exact height of the room.

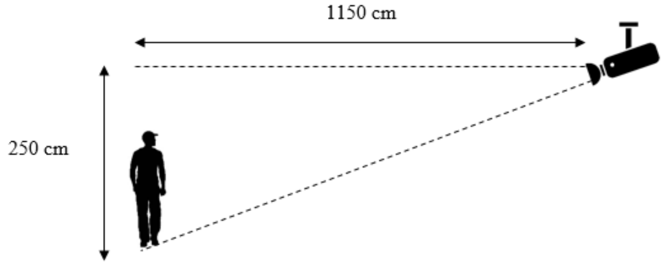

Fig. 11. Setup for testing

From these parameters, the pixel density was obtained using the equation:

$$pixel\ density = \frac{pixels}{height}$$

The camera used has a maximum height of 1080 pixels, therefore the equation will be

$$pixel\ density = \frac{1080\ \text{pixels}}{250\ \text{cm}}$$

$$pixel\ density\ = 4.32\ \text{ppcm}$$

The accuracy of the project was tested by comparing the theoretical value of a person's height in pixels with the actual detected pixels on the program. The theoretical value was obtained using the equation:

$$height\ in\ pixels = (height\ in\ cm)(pixel\ density)$$

If the height of the person is 165.1 cm (5 ft. 5 in.) tall, the equation will be

$$height\ in\ pixels = (165.1\ \text{cm})(4.32\ \text{ppcm})$$

$$height\ in\ pixels = 713.232\ \text{pixels}$$

The percent difference is then obtained by using the equation

$$\%diff_{height} = \frac{|experimental\ value - theoretical\ value|}{theoretical\ value} \times 100\%$$

For example, if the obtained value is 720.120 pixels, then the percent difference is

$$\%diff_{height} = \frac{|720.120 - 713.232|}{713.232} \times 100\%$$

$$= 0.9657\%$$

Then the accuracy is taken by subtracting the percent difference from 100

$$Accuracy_{height} = 100 - \%diff_{height}$$

$$Accuracy_{height} = 100 - 0.9657\%$$

$$Accuracy_{height} = 99.0343\%$$

The average accuracy is then obtained by adding all the accuracies obtained and dividing by the total number of samples.

$$Average\ Accuracy_{height} = \frac{\sum_{i=1}^{N}(Accuracy_{height})}{Total\ number\ of\ samples}$$

The response time for each trial is also taken, and it must be close to the expected 35 ms response, based on a camera that has 29 frames per second. The percent difference is obtained by the equation

$$\%diff_{response} = \frac{|response\ time\ in\ ms - 35\ \text{ms}|}{35\ \text{ms}} \times 100\%$$

And then the accuracy of the response is solved.

$$Accuracy_{response} = 100 - \%diff_{response}$$

For the statistical analysis, the standard deviation was computed to determine the average difference between each of the tests. The equation used for obtaining the standard deviation is

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x-\overline{x})^2}{N}}$$

and the mean is obtained by

$$\overline{x} = \frac{\sum_{i=1}^{N}(x)}{N}$$

where

s – standard deviation
x – detected value
x̄ – mean
N – total number of samples

Table I shows the summarized results of testing on different people with different heights at a fixed distance of 1150 cm away from the camera, all doing the same gesture by taking one step forward to ensure repeatability.

TABLE I. DATA TABLE FOR HEIGHT MEASUREMENT

Height in Pixels	Computed Value		Detected Value	
	Frequency	Percentage	Frequency	Percentage
578.79 - 652.09	3	10	7	23.33
652.2 - 725.60	21	70	8	26.67
725.61 - 799.01	6	20	9	30
799.02 - 872.42	0	0	5	16.67
872.43 - 945.83	0	0	1	3.33

Mean: 725.614
Standard Deviation: 73.4143

Table II shows the summarized results of the response time between the movement of the person and the recording of the video. These tests were done simultaneously with the height measurement.

TABLE II. DATA TABLE FOR RESPONSE TIME MEASUREMENT

Response Time in milliseconds	Detected Value	
	Frequency	Percentage
26.44 - 30.41	6	20
30.42 - 34.39	10	33.33
34.40 - 38.37	7	23.33
38.38 - 42.37	7	23.33

Mean: 34.4
Standard Deviation: 3.98

Table III and Table IV show the range of accuracies for both the height and response time measurement.

TABLE III. DATA TABLE FOR ACCURACY OF HEIGHT MEASUREMENT

Accuracy	Frequency	Percentage
80 - 85	6	20
86 - 90	7	23.33
91 - 95	8	26.67
96 - 100	9	30

Average Accuracy: 91.137021%

TABLE IV. DATA TABLE FOR ACCURACY OF RESPONSE TIME MEASUREMENT

Accuracy	Frequency	Percentage
80 - 85	6	20
86 - 90	8	26.67
91 - 95	7	23.33
96 - 100	9	30

Average Accuracy: 90.285714%

IV. CONCLUSION AND RECOMMENDATION

From the data gathered during testing, the researchers observed that there is very little difference in the theoretical and actual values. Although there are times that the actual values have a large difference from the theoretical values, it rarely exceeds a 15 percent difference, and the resulting average still exceeds expected 85 percent accuracy. The height of the person does not affect the accuracy, and it is noticeable that detected pixels are usually have larger values than the theoretical values since motion tends to add pixels on the screen. There is also a minor delay between the detection of motion and recording, but it is almost negligible since the delay can only be noticed through monitoring it through the block diagram by using timers. The delay values are very close to 35ms, which is the duration between each captured frame in a 29 fps camera. This means that the delay mostly depends on the camera used and not the program itself. With all the general and specific objectives met, the proponents can now conclude that the finished project is a security system that efficiently uses system memory for recording, that can properly detect the height of moving objects and has a fast response time.

The researchers suggest the following to those who plan to improve the project presented.

1. Additional features such as facial recognition, zooming in to the moving object, and any other possible features that can be done with image processing.
2. A network that uses multiple cameras, since this project uses only one camera.
3. Use of other cameras and microprocessors to increase performance and reliability.

ACKNOWLEDGMENT

First, we would like to thank God for giving us the strength and knowledge that we needed to make the project. Lord, we are always glad to have you as part of our success.

To Engr. Evelyn Q. Raguindin, for her guidance and patience in helping us through each part of writing a great paper.

To Engr. Mark Angelo C. Purio, for giving us the idea for this project, and for giving us technical advice whenever we encounter some problems.

To Engr. Edwin L. Astorga, for the help with a problem we encountered with the program, by providing us the necessary mathematical equations we needed for testing.

To Prof. Anna A. Ramos, for guiding us in doing the proper statistical analysis for this study.

And especially to our parents who were always there to provide financial support, love and care throughout the making of this thesis.

REFERENCES

[1] M. Cross, The Guardian. Explainer: CCTV, Available http://www.theguardian.com/commentisfree/libertycentral/2009/nov/06/explainer-cctv-surveillance-cameras, 2009.
[2] M. Bramberger, J. Brunner, & B. Rinner, Real-Time Video Analysis on an Embedded Smart Camera for Traffic Surveillance, IEEE, 2004
[3] M. Valera, & S. Velastin, Intelligent distributed surveillance systems•: a review, IEE, 152(2), 2005.
[4] E. C. Hannah, "Method and Apparatus for Processing Digital Video Camera Signals," US Patent 5 568 192, October 22, 1996.
[5] J. A. Kalomiros & J. Lygouras, Design and evaluation of a hardware / software FPGA-based system for fast image processing, 32, 95–106, 2007
[6] A. Bovik, The Essential Guide to Image Processing, 2009.
[7] T. Klinger, Image Processing with LabVIEW and IMAQ Vision, 2003
[8] D. Chun, "CCTV System Having Improved Detection Function and Detecting Method Suited for the System," US Patent 5 671 009, September 23, 1997.
[9] D. Tian, "Utilizing a Smart Camera System for Immersive Telepresence," US Patent Application 2014/0253667 A1, September 11, 2014
[10] N. Cohen, J. Gattuso, & K. MacLennan-Brown, CCTV Operational Requirements Manual, 2009.