Real-time Surveillance System Using Pattern Matching

Rahmadya Trias Handayanto, Retno N. Whidhiasih, Dadan Irwan
Computer Engineering
Islam 45 University
Bekasi, Indonesia
rahmadya_handayanto@unismabekasi.ac.id,
retno.nw@gmail.com, dadanirwan@gmail.com

Herlawati Herlawati
Information Engineering
STMIK Nusa Mandiri
Jakarta, Indonesia
Herlawati@nusamandiri.ac.id

Abstract—**Police department in every country has been trying to decrease crime's number, but in fact, the crime's number always grows every year. Instead of catching the criminal, we can minimize the crime by lowering the opportunity of criminal's action, so we propose the system to detect if there is someone entering the room by comparing pattern every couple of time. There are many pattern matching techniques, but the simple one is by counting its distance using Normal Euclidean. If there is a different pattern from previous pattern, the system will create an alarm. For this purpose, we use a capture application whose ability to capture an image periodically and the system that compare every image from previous capture. Testing result showed that our proposed system could detect every suspect object entering the area of our webcam or Closed-circuit Television (CCTV).**

Keywords: Surveillance; Pattern Matching; Normal Euclidean.

I. Introduction

We can find a lot of papers about research on surveillance system. They usually use capturing device such as webcam, Closed-circuit Television (CCTV), and so on, with some complex algorithms that can recognize some patterns. Because we only want to detect a suspect to our rooms, we must chose as simple as possible that can serve our needs[1].

We have read some research about surveillance. They are usually based on video, for example how to make good compression of data, such as Joint Photographic Experts Group (JPEG) 2000 [1]. This compression very important because the system use data communication when transfer the video to the storage which is need a lot of memory usage.

Some research about surveillance usually tracking an object. When someone enters an area being recorded, the system could track it [2, 6]. The pointer will follow where an object move is. This system could detect suspect objects with some difficult algorithms that usually need many resources. Because we only want to help the security especially with a lot of number cams in one screen, we propose the surveillance system with easy algorithm, so it can be implemented by low cost software and hardware.

Our system was inspired by previous research about detecting condition of our hotspot by interpreting network graph using neural network algorithm [3]. Instead of using neural network our system only compares new captured image to the previous one after capturing from webcam or CCTV.

We did not need to record all of our video from webcam because it will need a lot of memories but we only captured sample of pictures from webcams. The algorithm we have chosen also very simple. We compare new pattern from previous one using the Normal Euclidean method. It counts the distance of a pattern which is new capture from previous capture. If there is a distance, the system will create an alarm. Big numbers of distance means there are many different. It may be from someone entering the room or there is something stolen.

This paper was organized into some chapters. After discussing about foundation theory, the description of our system are investigated, and followed by testing and conclusion. The prototype we used was created using Matlab as a programming language and also can run by other similar languages that have open source characteristic like octave, scilab, and so on. We also use an automatic capture device that available in the market.

II. Basic Theory

Feature extraction process is needed if we want to gain the difference of our object from the other, but too many processes also need a lot of resources from our device. Because our propose system do some computation process periodically, we only use digital image processing to the captured object. This process can also make it easier to be processed by the next process.

A. Digital Image Processing

Captured images have Joint Photographic Group (JPG) or Bitmap Picture (BMP) extension with Red Green Blue (RGB) format. In order to be processed, we must convert it into matrix or vector. The process to get matrix consist of some stages: RGB to Binary conversion, cleaning and resizing. This stage made our image became easy to process.

[1] Demonstration of this system can be seen at: http://www.youtube.com/watch?v=XAWRNFqsrdU&feature=plcp.

978-0-7695-4926-2/12 $26.00 © 2012 IEEE
DOI 10.1109/EMS.2012.58

B. Normal Euclidean

Normal Euclidean is a theory about counting the distance of some point to other point. If we compare two similar images, the distance must be near zero, so by using this theory we can determine two images different or not.

The distance of two points can be counted by using this equation [5].

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1} \qquad (1)$$

For matrix we compare every point of one matrix to the other matrix having the same location column and row in its matrix. In Matlab language, we only use simple script[4]:

$$norm(x_1\text{-}x_2) \qquad (2)$$

Both x_1 and x_2 are two image matrixes that we compared. If x_1 and x_2 exactly the same, the result must be zero or near zero.

III. Description of The System

We used webcam or other device like CCTV for capturing image before feeding it into our system. We do not need to buy an expensive webcam for this system because we only compare an image to previous image. The picture was caught using an application that capture a video every couple of time. Our storage capacity must be enough to handle it. Because we only notify if there is someone entering the room, we made an application which delete the image after processed by our application.

Images must be converted into binary matrix after being cleaned by image processing operation created by Matlab function. Because this operation have to run continuously after pressing the run button, we have created a simple script for iterate it continue before pressing the stop button. The digital image processing must run efficiently using simple algorithms.

We use the same folder as both for running program and storing images. For storage efficiency we delete periodically images after being processed by application because we only notice if there is a different image between current image to previously image. We used batch script (DOS command) for rename image before processing and also delete it after processing. In order to integrate it with Matlab, we must convert that batch file into executable.

```
ECHO OFF
REM  QBFC Project Options Begin
REM  Productversion:  0. 0. 0. 0
REM  Fileversion:  0. 0. 0. 0
REM  Internalname:
REM  Appicon:
REM  QBFC Project Options End
ECHO ON
cd d:\program
ren *.jpg capture.jpg
```

This batch file, after converting to executable then insert into Matlab script.

Fig 1 shows flowchart of our system. We have three separate applications (webcam, capturing system, and analyzer) and one storage.

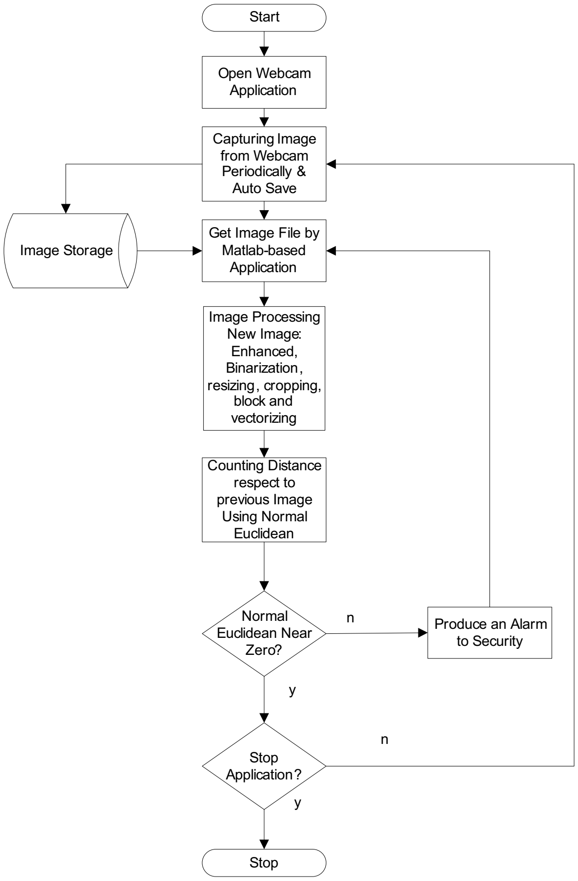
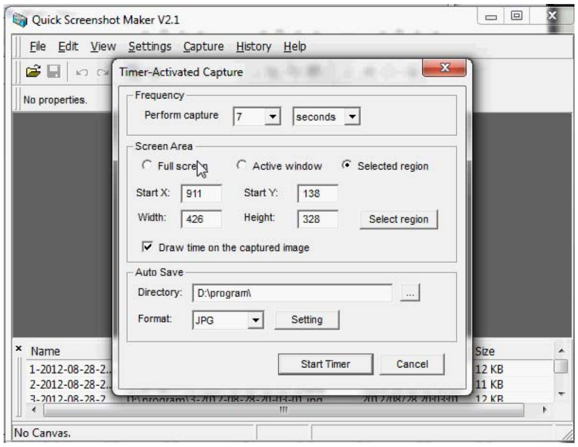

Start
Open Webcam Application
Capturing Image from Webcam Periodically & Auto Save
Image Storage
Get Image File by Matlab-based Application
Image Processing New Image: Enhanced, Binarization, resizing, cropping, block and vectorizing
Counting Distance respect to previous Image Using Normal Euclidean
Normal Euclidean Near Zero?
n
Produce an Alarm to Security
y
n
Stop Application?
y
Stop

Fig 1 The Flowchart of Our Proposed System

First, the webcam is setup and then the automatic capture software ready to capture some region on webcam. We set its timer faster then timer on application because we have a plan to delete an image after processing, so we make sure there always images to be processed. Second, after pressing the capture button the main application now already run after pressing the detect button. At last, our system will compare an image to the previous one periodically. If compare result with normal Euclidean is near zero the system understand that there is no suspect object entering the room. When the compare result show the distance is greater than zero, the system give a warning by producing an alarm, so the security will follow up this situation.

A. Preparation Stage

In this stage we must make sure our webcam/CCTV and capturing system run well. After region on automatic capture system has set up, we check if images were saved at proper location in our storage.



Quick Screenshot Maker V2.1
File Edit View Settings Capture History Help
No properties.
Timer-Activated Capture

Frequency
Perform capture 7 seconds

Screen Area
○ Full screen ○ Active window ● Selected region
Start X: 911 Start Y: 138
Width: 426 Height: 328 Select region
☑ Draw time on the captured image

Auto Save
Directory: D:\program\
Format: JPG Setting

Start Timer Cancel

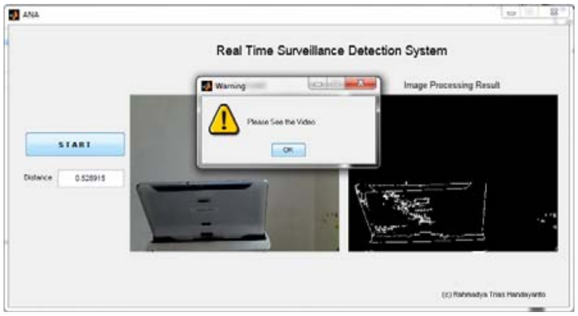
Name	Size
1-2012-08-28-2...	12 KB
2-2012-08-28-2...	11 KB
3-2012-08-28-2... D:\program\3-2012-08-28-20-03-01.jpg 2012/08/28 20:03:01	12 KB

No Canvas.

Fig 2 Example of Application for Capturing Periodically

B. *Analyzing Stage*

If preparation stage has done well, we come to our main system that analyze if there is a suspect or not. We created some scripts to get images captured by automatic capturing system periodically. After processing that image, the system will compare to the previous image. If there is a change the system will give a warning to security staff such as sound, warning message, sending short message service (SMS) and so on. Figure 3 shows an analyzer application created using Matlab language.



ANA
Real Time Surveillance Detection System
Image Processing Result
START
(c) Rahmadya Trias Handayanto

Warning
Please See the Video
OK

Distance 0.528915

Fig 3 Prototype of Our Proposed System

The image on the left side is original and on the right side is after image processing. we have matrix that only has number zero and one that represent background and line or point respectively.

We used some steps with image processing to get matrix of image. This matrix will be analyzed further by normal Euclidean. This script used Matlab function:

```
h=imread('capture.jpg');
delete('capture.jpg');
h2=rgb2gray(h);
h2=imadjust(h2);
y4=im2bw(h2);
y4=bwmorph(y4,'thicken');
y5=abs(y4-ones(size(y4)));
y6=bwmorph(y5,'clean');
y7=y6+zeros(size(y6));
```

Fig 4 shows image after being executed by this script.

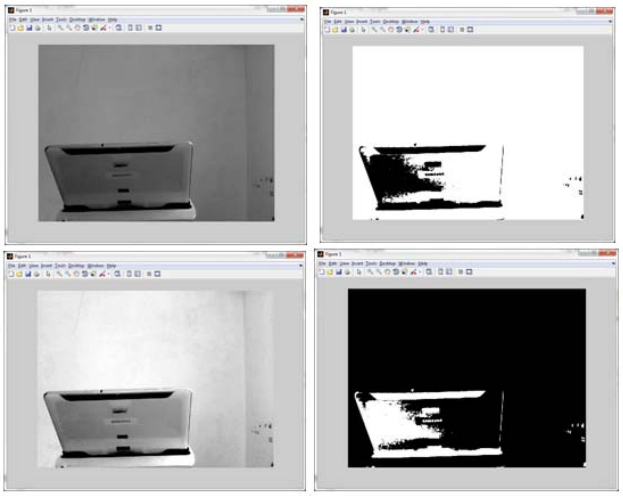


Fig 4 Image Processing for Getting Matrix

After counting the distance between current image to the previous one, we decide if the two images are similar or not. There usually noise that make two similar images being compared does not have a distance value zero, so we give a maximum distance 0.5 for similar image. If the distance counted by normal Euclidean is greater than 0.5 we will see a warning by message or sound. It means there is a suspect object that must be analyzed further by security staff. On the contrary, if the distance is smaller than 0.5, the system won't send a message. It means the room is secure.

C. *Improving Performance*

Although we can just compare two images, we can make our execution faster by using mean value of axis. We use this method because we only look for the different of one image to previous one.

For example, a 480 x 640 matrix become a vector with size only 480 x 1 when using this method. Because the input vector with size 1200 reduced to 20, it helps a computer counting the distance faster than before. If we have a matrix:

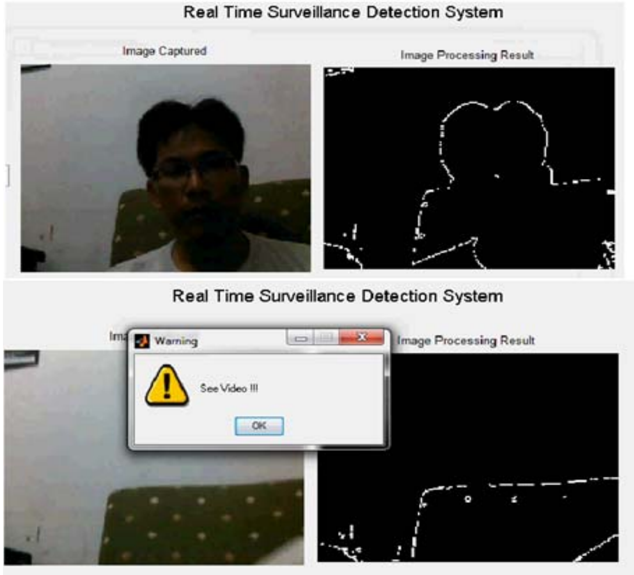
```
a =
    0.8147    0.6324    0.9575    0.9572    0.4218
    0.9058    0.0975    0.9649    0.4854    0.9157
    0.1270    0.2785    0.1576    0.8003    0.7922
    0.9134    0.5469    0.9706    0.1419    0.9595
    …
```

If we only two know the different from other matrix, we can just count the average of row:

```
Average =
    0.7567
    0.6739
    0.4311
    0.7064
    …
```

IV. Experimental Results

By comparing two similar images or there is no suspect object entering the room, we found that our system give result around 0.3, so we decide to give maximum distance 0.5 with addition 0.2 as a tolerance. Fig 5 shows when our system detect a change.


Real Time Surveillance Detection System
Image Captured
Image Processing Result
Real Time Surveillance Detection System
Warning
OK
Image Processing Result

See Video !!!

Fig 5 Warning Message after Detecting a Change

Testing result showed our propose system could give an alarm when there is a suspect object entering the room we are watching, but there are some noises that could give a bad result such as extreme different of lighting caused by turning of the light and the moving of webcam accidentally, and we think it also become a problem for another systems.

We also still had a problem related to delay of real video because our system needs time in analyzing images. However our system still understood whether there was a suspect object or not. However, this system can be implemented to our CCTV system for monitoring the car at parking area.

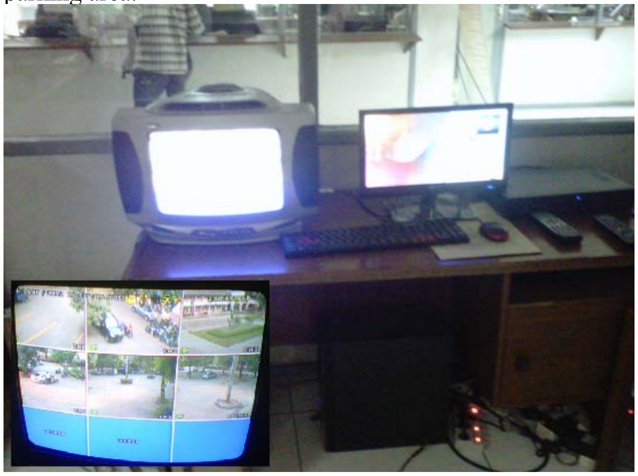


Fig 6 Implementation to Our Local CCTV for Car Parking

Table I shows the Euclidean distance of some suspect objects. There is a different between small and big object and also between empty room and room with stuffs. For empty room, our experiment showed the room with some stuffs always give smaller Euclidean distance number than the empty one.

TABLE I
SUSPECT OBJECT TESTING

No	Image	Euclidean Distance	Decision (Limit=0.5)	Euclidean Distance (Empty Room)	Decision (Limit=0.5)
1	No change	0.1	No Alarm	0.1	No Alarm
2	Lose something (e.g. laptop)	1.25	Warning	1.25	Warning
3	Small animal (e.g. Rat)	0.2	No Alarm	1.0	Warning
4	Big animal (e.g. cat)	0.6	Warning	1.03	Warning
5	Children	0.6	Warning	1.07	Warning
6	Man	0.7	Warning	1.1	Warning

V. COMPARISON TO OTHER SYSTEMS

It is easy to compare this system to other common systems because we compared captured image with the previous one after capturing them periodically. There are some research papers about tracking surveillance based on fuzzy or other statistical method [2, 6]. That system will track a suspect object but we think it will need a lot of resources if implementing in a large number of webcams/CCTVs.

Other system used fuzzy when implementing a computer vision technology with Adaptive Gaussian Mixture [6]. Also we can find Open Development Environment for Evaluation of Video Surveillance System (ODViS) for tracking the surveillance system [2]. It also the same as the previous one, except the method they are using and could not be implemented to our university because we have many CCTVs in every building that must be served.

We only proposed the system that can easily understand if there is a change in webcam's shoot. It can help security staff when monitoring a lot of rooms with CCTV or webcam devices. And every changing can be seen at the storage for being analyzed by the police later.

Our system and others that we have discussed also have difficulty in recognizing an object if there is no light in the night or the disturbance of its capturing device.

VI. Conclusions and Future Research

Testing result showed our system could detect if there is a change on area captured by webcam/CCTV. After detecting a change, system give a warning, so it will help security doing his job. For surveillance system having a lot of videos, this system can help securities in watching videos. Also for minimizing the use of storage, we can only save a change. By subtracting one captured video from previous, if there was a change we could captured suspect object.

If we subtract between two different images we will find suspect object. Therefore in the next research we will use this method to track it where it goes. We still have a problem of lagging between to images.

This application will be more useful if we use web-based webcam application because we can monitor our room from other place and with our propose system, we should not monitor our cam all the time, but when there is a suspect object entering the monitoring area, we will get notice such as sound, Short Message Service (SMS), email and so on.

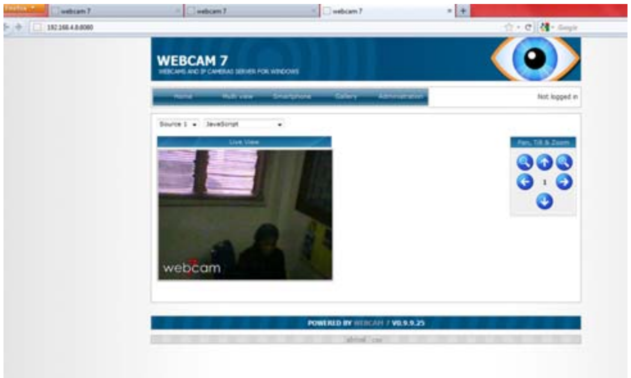


Fig 7 Web-based cam application that can be Integrated with our system.

There are some challenges that we will do for the next research from combining the alarm system with webcam/CCTV of our system, automatic turning on/off the lights, face recognizing, integrating with object tracking system, and so on. We must have a collaboration to other discipline for getting a perfect system.

Acknowledgment

We would like to thanks other people and institution who had helped this research. The Quick Screenshoot Maker (http://www.etrusoft.com) for it's trial version for capturing periodically the traffic's image and also the Matlab Software (http://www.mathworks.com) as a programming language for prototyping the system. The sounds of our application are recorded from http://www.translate.google.com, and finally we also thanks to Institute for Research and Community Services (LPPM) of Islam 45 University for funding this research.

References

[1] Hata, Toshihiko, Naoki Kuwahara, Toshiharu Nozawa, "*Surveillance System with Object-Aware Video Transcoder*". Journal of Mitshubishi Electric Research Laboratories. Massachusetts, 1-5. 2005.

[2] Jaynes, Christoper, Stephen Webb, R. Matt Steele, Quanren Xiong. (2001). "*An Open Development Environment for Evaluation of Video Surveillance Systems*", Journal Computer Science University of Kentucky, 1-8, 2001.

[3] Handayanto, Rahmadya Trias, Haryono, Jarot Prianggono, "*Real Time Neural Network-Based Network Analyzer for Hotspot Area*", Proceeding International Conference on Advance Computer Science and Information Systems (ICACSIS), 323-330. 2011.

[4] Gonzalez, Rafael C., Richard E. Woods, Steven L. Eddins, *" Digital Image Processing Using Matlab, Second Edition",* USA: Gatesmark Publishing, 2009.

[5] Leondes, Cornelius T. *"Image Processing and Pattern Recognition",* USA: Academic Press, 1998.

[6] Widyanto, M. Rahmat, "*Computer Vision Technology for human Surveillance Using Fuzzy Approach*", Proseeding International Seminar of Information Technology (ISIT), 7-16.