TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 14, No. 1, April 2015, pp. 123 ~ 129
DOI: 10.11591/telkomnika.v14i1.7257

Evaluation of Moving Object Detection Methods based on General Purpose Single Board Computer

Agung Nugroho Jati*, Ledya Novamizanti, Mirsa Bayu Prasetyo, Andy Ruhendy Putra
Department of Computer Engineering, School of Electrical Engineering, Telkom University,
Jl. Telekomunikasi Terusan Buah Batu, Bandung, Indonesia 40257, Ph. +6222-7564108
*Corresponding author, e-mail: agungnj@telkomuniversity.ac.id

Abstract

RGA and SKDA are two different methods which can be used to detect the object in image based processing. In order to support the moving surveillance camera system which proposed in Telkom University, RGA and SKDA have tested to be reviewed which more reliable to be implemented in a single board computer. In this paper, will be discussed about implementation and testing of two different methods of object detection using backgrounds subtraction. For implementation, each of them will be combined with Extended Kalman Filter in a Raspberry Pi. The parameter which have tested are memory and CPU usage, and system utilization. The result shows that RGA is more reliable than SKDA to implemented in SBC because of less CPU usage and system utilization.

***Keywords**: running gaussian average (RGA), sequential kernel density approximation (SKDA), extended kalman filter, single board computer (SBC), Raspberry Pi*

1. Introduction

The study of object detection based on image processing is the interesting way until this time, especially in the implementation. Many methods have modeled and developed before, tried to be implemented in many system. But there are many constraints in embedded system implementation. In an embedded system such as single board computer (SBC), there's lack processor with a limited memory capacity. Besides, limited power also will be a constraint in implementation.

Raspberry Pi as a SBC has a small size and compact for embedded system implementation [1]. As a compensation, it can't be upgraded by bigger memory capacity or higher processor. So, when someone try to implement a computing mechanism onto SBC, they must consider resources limitation. As we know, the implementation of signal processing needs much enough resources in a computer. So, in this paper will be evaluated an implementation using SBC. What can be happened when we move the complicated techniques to limited resourced computer.

Last year, proposed a moving surveillance camera in Telkom University for higher reliability in order to monitor indoor building [15]. Thus system consists of camera and SBC which can move follow the human movement. Consider the limitation of SBC, must be found best method with high accuray but still managed by a small size memory. Besides, the processing time still can be tolerated.

There are two kind of main process in the system. The first is object detection. Many methods are developed in this way. Most of them use background substraction techniques, such as Running Gaussian Average, Temporal Median Filter, Mixture of Gaussian, Kernel Density Estimation, Sequential Kernel Density Approximation, and many more [2-4]. in this paper, will be only discussed about RGA and SKDA.

We only tried to implement RGA and SKDA and evaluate them because RGA is said as a simple algorithm with an acceptable accuracy and only needs less memory consumption [5]. SKDA, is more complex algorithm but has high accuracy but still needs lower time complexity to be processed [6]. In the end, you can see the conclusion, where is fitter than other to be implemented in SBC.

Received December 29, 2014; Revised February 20, 2015; Accepted March 15, 2015

The second process is movement prediction. Extended Kalman Filter (EKF) is used as an algorithm for predicting the movement [7]. The output from SKDA or RGA, processed in EKF to produce the direction. And then, SBC will command the camera to move directly to the object. The presentation is organized as follow: in the second section, will be described the method which is implemented in SBC. In the third section, we'll discuss about the result of implementation, given the analysis and evaluation. And the last, conclusions are given.

2. Research Method

2.1. Running Gaussian Average (RGA)

Running Gaussian Average (RGA) is one of many methods in bacground substraction [8]. This method can be used to detect the object, static or dynamic. RGA is also called as 1-G (One Gaussian) [2]. In RGA, background pixel modeled as Gaussian Distribution (μ,∑), where:

$\mu_{s,t}$: mean point of each pixel and time;

$\sum_{s,t}$: covariance matrix of each pixel and time [5].

Covariance matrix values depend on noise which is contained by the image. Higher noise will produces higher covariance matrix too. In the other words, higher noise in the image will also produce higher temporal gradient $|I_{s,t} - \mu_{s,t}|$, so the pixel can be said as moving pixel.

The mean and covariance are dynamic values, and can be updated by the formulas:

$$\mu_{s,t+1} = (1-\alpha).\mu_{s,t} + \alpha . I_{s,t} \quad (1)$$

$$\sum_{s,t+1} = (1-\alpha).\sum_{s,t} + \alpha.(I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T \quad (2)$$

In SBC implementation, covariance matrix can be optimized by calculate only the diagonal of the matrix. Because in general, covariance matrix consist of 3x3 matrix. Besides, can be used spatial agregation as like as morphological filter to make better performance of background substraction technique [2].

2.2. Sequential Kernel Density Approximation

In Sequential Kernel Density Approximation (SKDA) method, mean-shift vector is used to track the gradient which can detect main modes from pdf (probability density function) time from data sample by minimal set assumption directly [8]. SKDA uses Gaussian Kernel, noted by x_i (i = 1,…,n), where:

x_i : Gaussian mean value;

π : variance of covariance dxd related with Gaussian [4].

The finction from density point define as:

$$f(x) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n} \frac{k_i}{|P_i|^{1/2}} \exp(-\frac{1}{2} D^2(x, x_i, P_i)) \quad (3)$$

Where,

$$D^2(x, x_i, P_i) \equiv (x - x_i)^T P_i^{-1} (x - x_i) \quad (4)$$

SKDA is optimized from Kernel Density Estimation (KDE). Where sample frame taken process is compressed so time process in more efficient. This is happened because less frame is sent to be processed, so the computation is lees complexity [8].

2.3. Extended Kalman Filter

Extended Kalman Filter (EKF) is a set of functions which are used to estimate current system state and correct them byased on previous state. EKF uses two calculation parameter, they are prior state and posterior state [7, 9]. You can see as follows:

Prior state calculation given as:

$$x = f(x_t, x_{t-1}) + \underbrace{\frac{\partial_{f}(x_t, x_{t-1})}{\partial_{\mu_{t-1}}}}_{F_t}(x_t - x_{t-1}) \quad (5)$$

Posterior State Calculation given as:

$$x = x + K(z_t - h(x_t)) \quad (6)$$

Where K is Kalman gain given below:

$$K = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \quad (7)$$

And h(X) is a calculation function as follows:

$$h(x_t) = h(x_t) + \underbrace{\frac{\partial_{h}(x_t)}{\partial_{\mu_t}}}_{H_t}(x_{t-1} - \bar{x}_t) \quad (8)$$

∑: covariance error, where can be calculate by:

$$\Sigma_t = F_t \Sigma_{t-1} F_t^T + \mathrm{R}_t \quad (9)$$

$$\Sigma_t = (I - \mathrm{K}_t \mathrm{H}_t)\Sigma_t \quad (10)$$

H & *F*: calculation matrix, calculate by Jacobian below:

$$\frac{\partial h(x_t, x_{t-1})}{\partial x_{t-1}} \text{ for } H \quad (11)$$

$$\frac{\partial f(x_t, x_{t-1})}{\partial x_{t-1}} \text{ for } F \quad (12)$$

Q: calculation error.

From those formulas, *I* is defined as identity matrix and R is defined as calculation error. Dynamic system can be taken by replacing value of z_t be x_{t-1} . Those technique optimizes the direction of camera detection when there are more than one object exist.

2.4. Object Movement Model

Object movement is modeled by a linear movement. Gotten by calculating the difference of current frame coordinate and previous one. So, the model given as follows:

$$x_t = x_{t-1} + \Delta x_{t,t-1}; \quad (13)$$

$$y_t = y_{t-1} + \Delta y_{t,t-1}; \quad (14)$$

Where *x*, is a horizontal axis, y is vertical axis, and t is the time.

2.5. Raspberry Pi Model B

As a SBC, we use Raspberry Pi model B. Raspberry Pi uses SoC from Broadcomm BCM2835 and consist of ARM11 700 MHz processor, 512 MB RAM, and GPU Videocore IV. This SBC uses SDCard for booting sequence and storage system [10, 11].

2.6. Evaluation Criteria

There are two kind of evaluation criterias. First criteria is related with each background substraction methods. For RGA, the parameters are learning rate (a) and deviation standart constant (k) [5, 8]. And in SKDA, they are data variances (σ) and *t* (*threshold*) [2, 12]. Second criteria, which are wanted to know to see performances of implementation in SBC. They are including memory and CPU usage, and system utilization.

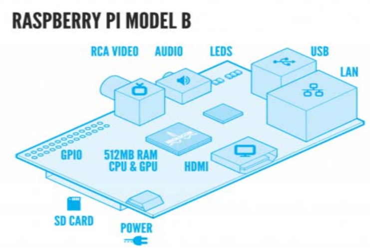


Figure 1. Raspberry Pi model B [8]

3. Results and Analysis

3.1. System Design

System hardware consist of camera, SBC Raspberry Pi, and servo motor for directing camera to the object. First, system has to detect the object by RGA or SKDA. If there are any movement of the object, system will follow that movement. In order to smooth the system movement, Extended Kalman Filter is implemented. In the simple way, system described by this picture below.

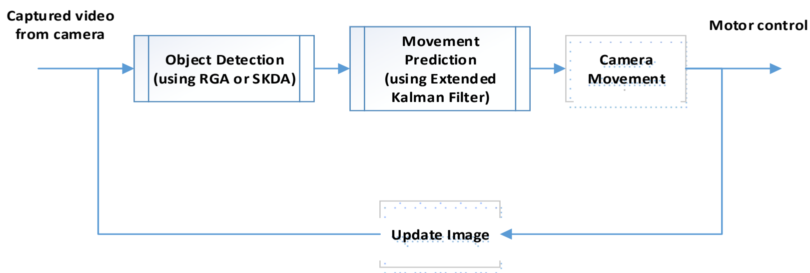


Figure 2. System Implementation Scheme

3.2. RGA and SKDA Parameters Analysis

As explained before, SKDA has two important parameters. They are variance (σ) and threshold (t). We've already tested implementation of SKDA as a background substraction in some of scenarios. The results are described below.

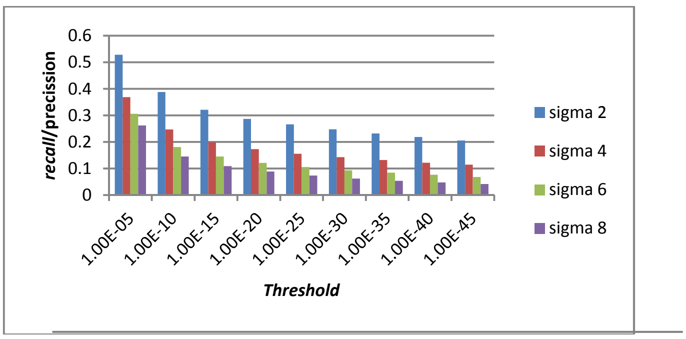


Figure 2. SKDA Precision Values Depends on Threshold and Variance

Based on the test, optimal value of the precision of the system are happened when variance 2 and threshold 1e-5. So, that value will be used for the implementation and performance testing.

RGA also uses 2 parameters, they are learning rate and threshold. We've try to change the values of parameters in order to find optimal system. The result is described as follows.

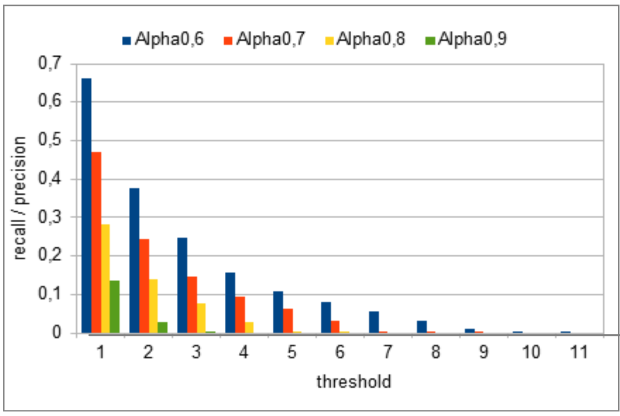
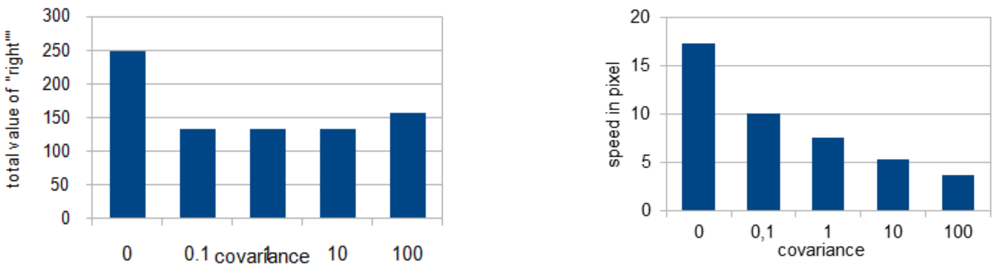


Figure 3. RGA Precision Values Depends on Learning Rate and Deviation Standart

Optimal precision is achieved when a set as 0.6 and t 1. Based on the graphic, higher threshold will make precision less and become un-optimal. This is also happened for the higher a value.

3.3. EKF Parameter Testing

EKF is used to predict the next movement of object. EKF performance depends on the Q (covariance) parameter. In this research, we've tried some variances of Q value and the result is described in the figure below.



(a) (b)

Figure 4. (a) amount of true prediction; (b) "speed" of prediction

From the graphics above, shown that higher covariance will impact the "speed" of prediction to become less. For implementation, Q is set as 0.1 cause it has highest "speed" and still tolerated "true" prediction. Q=0 can't be used because will give many "jump" processes.

3.4. SBC Perfomance Analysis

SBC based system performance can be seen by analyzing the memory and CPU usage [13, 14], also the system utilization. This test is proposed to know which technique is more compatible to be implemented in SBC.

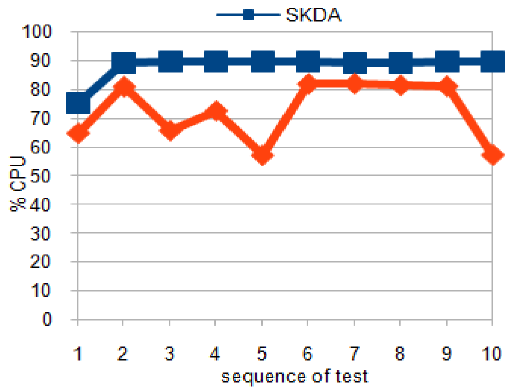


Figure 5. RGA and SKDA Computation

Based on the figure above, shown that SKDA use arround 10% CPU resource more than RGA. RGA uses CPU arround 73.45% average when is executed. And SKDA, use average 88.24% during executed by the system. Despite RGA has less usage of CPU, it uses large amount of memory capacity during the process. It's described in the figure below.

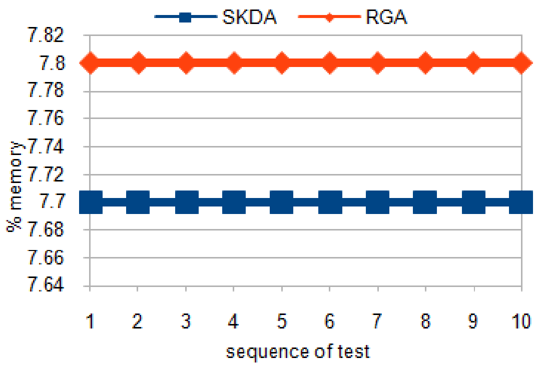
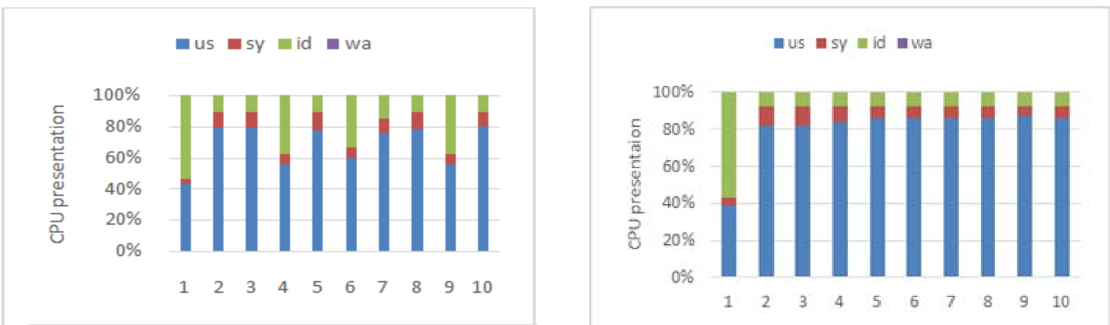


Figure 6. RGA and SKDA Memory Usage

RGA uses more memory resource because it needs more memory space for deviation standart of each pixel (used 2 frames of every process). While SKDA only uses memory for putting 2 frames in every process.



(a) (b)

Figure 7. System Utilization (a) RGA; (b) SKDA

From Figure 7 above, we can say that RGA is more "healthy" than SKDA. The system is always in high utilization when system implement SKDA as background substraction technique. SKDA has less user time but higher idle time of the processor than RGA. It's happened because by using SKDA as a backgound substraction technique, system more often send request data to the hardware level.

4. Conclusion

Based on the implementation, testing, and analysis can constructed some conclusions of the study. As a limited resources computer, single board computer can't process a high level signal processing as realtime. There have been tested two kind of background substraction technique combined by EKF in a SBC. Both RGA and SKDA can be used as a background substration technique to detect the moving object in a SBC. But it's required more than 256 MB memory capacity. RGA indeed, needs more memory usage (arround 0.1%) than SKDA to store deviation of each pixel for every frame. In the implementation, used 2 frames of captured video.

However, RGA uses less resource of CPU in the process than SKDA. RGA just needs in average 73.45% of CPU resource while SKDA needs in average 88.24%. Because of that, SKDA needs more time processing. And it also influence the system utilization. Based on the study, known that SKDA will make system always in high utilization.

All of the results of this study shown that RGA is more reliable to be implemented in SBC, especially Raspberry Pi. Still and all, the processing time needs to be increased more. It can be happened by replace the SBC with better CPU speed and higher memory capacity.

In the further stage of study, the result from this study can be expanded to try other background substraction methods. And also in thext researches, can be implemented directly as a surveilance camera to follow the human movement.

References

[1] The Rapsberry Pi Foundation: http://www.raspberrypi.org/. October 30th, 2013.
[2] Benezeth Y, Jodoin PM, Emile B, Laurent H, Rosenberger C. Comparative Study of Background Subtraction Algorithm. *Journal of Electronic Imaging*. 2010: 1-30.
[3] Tsai DM, Lai SC. Independent Component Analysis-Based Background Subtraction for Indoor Surveillance. *IEEE Transaction on Image Processing.* 2009; 18(1): 158-167.
[4] Maddalena L, Petrosino A. A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. *IEEE Transaction on Image Processing.* 2008; 17(7): 1168-1177.
[5] C Wren, A Azarbayejani, T Darell, AP Pentland. Pfinder: Real Time Tracking of the Human Body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 1997; 19(7): 780-785.
[6] B Han DC. *Sequential Kemel Density Approximation through Mode Propagation: Applications to Background Modeling*. Proceedings of Asian Conference .2004.
[7] Huang S. Understanding Extended Kalman Filter - Part III: Extended Kalman Filter. 2010.
[8] Piccardi M. *Background SubtractionTechniques: a Review*. IEEE International Conference on Systems, Man and Cybernatics. 2004: 3099-3104.
[9] Ribeiron M I. Kalman and Extended Kalman Filter: Concept, Derivation, and Properties. 2004.
[10] The Rapsberry Pi Foundation. *FAQs | Rapsberry Pi*. October 30th, 2013. Rapsberrypi.org: http://www.raspberrypi.org/faqs.
[11] The Raspberry Pi Foundation. *About us | Raspberry Pi*. October 30th, 2013. Raspberry pi org: http://www.raspberrypi.org/about.
[12] Urbach ER, Wilkinson MH. Efficient 2-D Grayscale Morphological Transformations With Arbitrary Flat Structuring Elements. *IEEE Transaction on Image Processing.* 2008; 17(1): 1-7.
[13] IBM. Linux Performance and Tuning Guidelines. New York: IBM Group. 2004.
[14] StrongMail System. Extreme Linux Performance Tuning and Monitoring. StrongMail System.
[15] Andy Ruhendy, Agung NJ, Ledya N. *Sistem Moving Surveillance Camera Berbasis Sequential Kernel Density Approximation dan Extended Kalman Filter*. Seminar Nasional Teknologi Terapan. 2014: 304-307.