PARTICLE FILTER WITH BINARY GAUSSIAN WEIGHTING AND SUPPORT VECTOR MACHINE FOR HUMAN POSE INTERPRETATION

Indah Agustien[1,2*)], M. Rahmat Widyanto[1], Sukmawati Nur Endah[1,3], and T. Basaruddin[1]

1. Faculty of Computer Science, University of Indonesia, Depok 16424, Indonesia
2. Faculty of Engineering, University of Trunojoyo, Bangkalan, Madura 16912, Indonesia
3. Faculty of Mathematics and Natural Sciences, Diponegoro University, Tembalang, Semarang 50239, Indonesia

*)E-mail: indah.agustien@if.trunojoyo.ac.id

Abstract

This paper proposes human pose interpretation using particle filter (PF) with Binary Gaussian Weighting and support vector machine (SVM). In the proposed system, particle filter is used to track a human object, then this human object is skeletonized using thinning algorithm and classified using SVM. The classification is to identify human pose, whether it is normal or abnormal behavior. Here particle filter is modified through weight calculation using Gaussian distribution to reduce the computational time. The modified particle filter consists of four main phases. First, particles are generated to predict target's location. Second, the weight of certain particles is calculated and these particles are used to build Gaussian distribution. Third, the weight of all particles is calculated based on Gaussian distribution. Fourth, particles are updated based on each weight. The modified particle filter could reduce computational time of object tracking since this method does not have to calculate particle's weight one by one. To calculate weight, the proposed method builds Gaussian distribution and calculates particle's weight using this distribution. Through an experiment using video data taken in front of the cashier of a convenience store, the proposed method reduced computational time in tracking process until 68.34% in average compared to the conventional one, meanwhile the accuracy of tracking with this new method is comparable with particle filter method, i.e. 90.3%. Combining particle filter with binary Gaussian weighting and support vector machine is promising for advanced early crime scene investigation.

Keywords: Particle filter, prediction, skeletonization, support vector machine, update

1. Introduction

Currently, the number of criminal actions is increased and the modus of actions varied [1]. Therefore, police department in Jakarta obligated all office buildings, meeting rooms, mall and other public places to be equipped with circuit closed television (CCTV) [2]. Hence security officer could monitor all incidents that happen in those places from video data taken from CCTV. Unfortunately, officer could not always keep an eye on those monitors from CCTV when they are tired, sleepy and etc. Consequently, criminal actions could not be avoidable.

Therefore human pose interpretation system to process video data is proposed in this research, in order to give early warning of criminal actions to the security officer. There are three subsystems in human pose interpretation system, i.e. human object tracking using modified particle filter, skeletonization using thinning algorithm and classification using SVM.

The remainder of this paper is organized as follows. Section 2 is methods, it describes system design of human interpretation. Section 3 is result and discussion and final section is conclusions of this research.

2. Methods

Human pose interpretation system (HPIS) gives interpretation of human pose from video data. The objective of this system is to distinguish human pose whether it is normal or abnormal behavior. The Design of this system can be seen in Figure 1.

HPIS is divided into three main subsystems. First, human tracking using modified PF to track human's location in each frame from video data. Second, skeletonization using thinning algorithm to obtain tracked human's skeleton. Third, classification using SVM to classify human pose, whether it is normal or abnormal behavior.

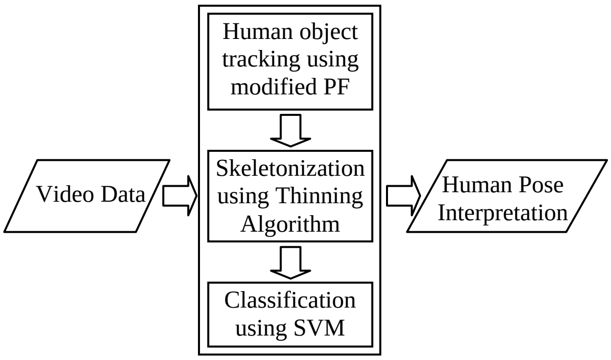

Figure 1. Human Pose Interpretation System

The first subsystem is human tracking using modified PF. PF needs human's detection to detect human's location in the first frame. Then this detected location is used to track human's location in the remainder frames. Human detection developed by Laptev [3] is used in the proposed system.

The Second subsystem is skeletonization using Thinning algorithm. Skeletonization is the process of making the skeleton image using thinning algorithm. In HPIS, an iterative parallel algorithm which presented by Zhang and Sue [4] is used to obtain the skeleton image.

The third subsystem is classification using SVM. SVM is used due to its high recognition accuracy [5]. The objective of this subsystem is to classify whether tracked human object is normal or abnormal behavior. Following subsections give detail of each subsystem design.

Human Object Tracking Using Modified PF. PF was first suggested by Gordon et.al in 1993 [6]. This method is derived from Bayesian approach. PF is used in the tracking subsystem since it is a robust method in noisy image [7-10]. This method consists of two primary stages, i.e. prediction and update [6-14]. In the prediction stage, numbers of particles are generated to predict target's location in the current frame from video data. In the update stage, these predicted particles are updated based on each weight.

Since particles are representation of prediction of the actual target's position in current time, therefore the more particles generated, the more predictions are made. Hence this particle filter method can achieve a high accuracy as the number of particle is increased. However, as the number of particle increases then the computational time is also increased [15]. For that reason, new method is developed to solve disadvantage of particle filter.

Before particle filter method is modified, first characteristic of particle filter is observed. Therefore modified PF will have tracking accuracy as good as conventional PF, but does not need a huge computational time of tracking process. The observation is done particularly in the filtering stage, especially in the weight's calculation process, since this process needs 60-80% of computational time in average [15]. Hence new method is focused in the weight's calculation.

Particle's weight value represents how close particle to the actual target's location. It means that particle will have large weight value if it is close to the actual target's location. On the contrary, it will have small weight value if it is far from the actual target's location. This characteristic is similar with Gaussian distribution as depicted in Fig. 2.

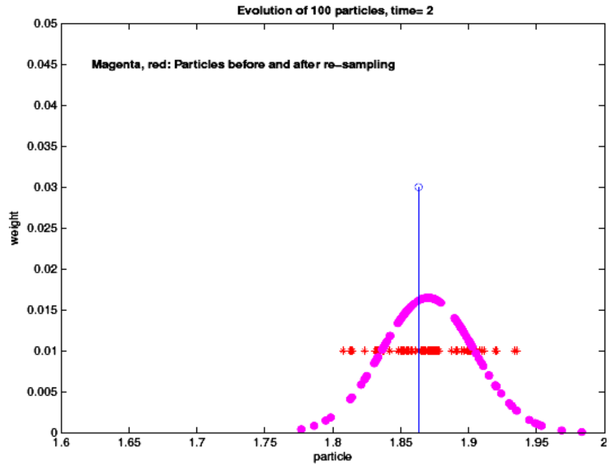

Figure 2. Weight Value Distribution [16]

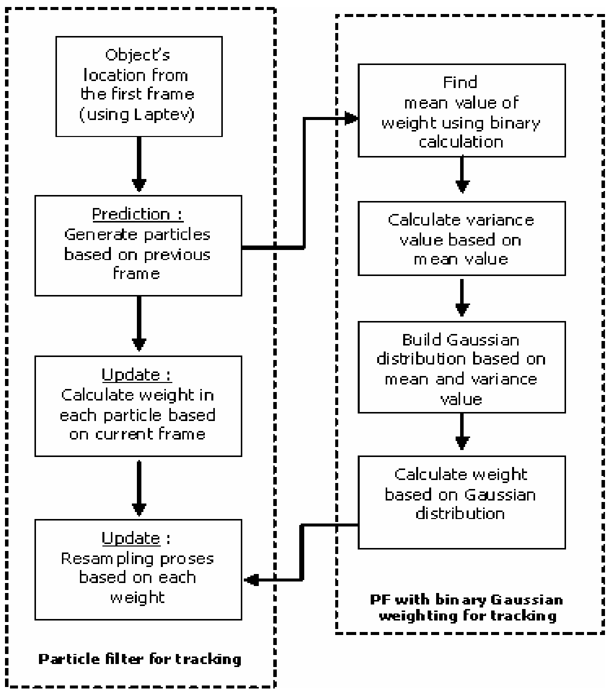

Figure 3. Conventional PF and Modified PF (PF with Binary Gaussian Weighting)

Here is binary calculation to find mean value.

```
BinaryCalculation
{
      middleParticle=(beginParticle+endParticle)/2;
      mid1= (beginParticle+middleParticle)/2;
      mid2=(middleParticle+endParticle)/2;
      weightMiddle=weightCalc (middleParticle);
      weightMid1=weightCalc (mid1);
      weightMid2=weightCalc(mid2);
      IndMax=max(weightMid1,weightMiddle,weightMid2);
      switch IndMax
            case {1}
                        beginParticle= beginParticle;
                        endParticle=middleParticle;
            case{2}
                        beginParticle=mid1;
                        endParticle=mid2;
            case{3}
                        beginParticle=middleParticle;
                        endParticle=endParticle;
      end
}
```

Second, calculate variance value based on mean value. Mean value from binary calculation is used to calculate full width half maximum (FWHM) as depicted in Figure 4.

Variance value is calculated from FWHM value using equation 1,

$$FWHM = 2.355 \times \sigma \qquad (1)$$

Third, build Gaussian distribution based on mean and variance value. After all parameters of Gaussian are calculated, then build Gaussian distribution using Equation 2 as follows,

$$y = ae^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (2)$$

where a, μ, σ is height, mean and variance of Gaussian distribution respectively.

Fourth, calculate weight based on Gaussian distribution. Using Equation 2, weight of all particles is calculated. Hence weights are not calculated one by one. Hopefully, this new method could reduce computational time especially in weight calculation.

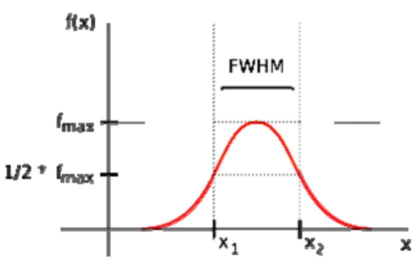

Figure 4. FWHM and Mean Value [17]

Skeletonization Using Thinning Algorithm. The second subsystem is skeletonization using thinning algorithm. To achieve skeleton image, thinning algorithm is used. The input is RGB images with size of 64 x 128 pixels. The RGB images are converted into binary images where the human pose is the connected white pixels. Then the binary images are used for the thinning process. An iterative parallel algorithm which presented by Zhang and Sue [4] is used in the thinning algorithm.

The pseudo code of the algorithm uses the following variables. I = the original binary image in which black pixels are “0” and white pixels are “1”. The object in the image is made of connected white pixels. J and K are temporary images used in iterations of the algorithm. J is the $(n\text{-}1)^{th}$ iteration output and K is the current, or nth, iteration output. P(i) is the current pixel under consideration. The 8-neighborhood around the pixel P1 is shown Fig. 5.

Two other variables used in the algorithm are A and B. A of pixel P1 is defined as: SUM (P2…P9). B of pixel P1 is the count of the number of 0 to 1 transitions in a

P9 (i-1,j-1)	P2 (i-1,j)	P3 (i-1,j+1)
P8 (i,j-1)	P1 (i,j)	P4 (i,j+1)
P7 (i+1,j-1)	P6 (i+1,j)	P5 (i+1,j+1)

Figure 5. The 8-Neighborhood of Pixel P1

0	0	1
0	P1 (i,j)	1
1	0	1

Figure 6. Calculation of B

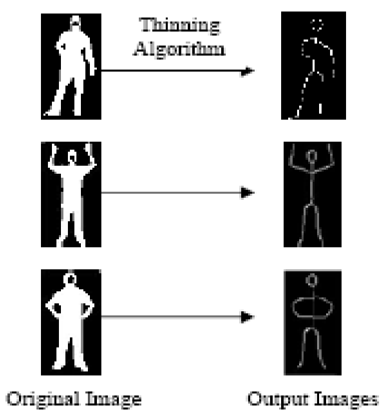

Figure 7. Original and Output Images Using Thinning Algorithm

clockwise circle from P9 back to itself. For example suppose we have the pixels shown in Fig. 6. B would equal 2 since there are two 0 to 1 transitions in a clockwise order.

The algorithm runs in 2 sub-iterations. During each sub-iteration, different rules are applied for deciding whether or not a pixel will be deleted. The outputs of this algorithm on some binary images are shown in Fig. 7. The value of every pixel of the skeleton image is used for the classification phase. For efficiency, the skeleton images are resized by a half of the original images where its dimension is 32 x 64 pixels.

Classification Using SVM. The third subsystem is classification using SVM. Input to this SVM is skeleton image from second subsystem. Classification subsystem is a system for determining a human pose belonging to normal or abnormal behavior. Normal pose means that the person will not do any criminal actions, while abnormal is the opposite. The pose is assumed normal if it is a standing pose with the two hands lay beside human body, conversely abnormal pose if two hands or one hand on top of the body. To determine this, a threshold line is defined as a line as long as two hands like Fig.8. The pose of sitting human include abnormal pose.

The classification involves with the training and testing data which consists of some data instances can be seen in Fig. 9.

Each instance in the training set contains one “target value" (class labels) and several “attributes" (features). The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

Given a training set of instance-label pairs (x_i, y_i), $i = 1, 2, ...l$ where $x_i \in R^n$ and $y \in \{-1,1\}^l$, the support vector machines (SVM) [19-20] require the solution of the following optimization problem:

$$\min_{w,b,\varepsilon} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \varepsilon_i \qquad (3)$$

$$\text{subject to} \quad y_i (w^T \phi(xi) + b) \geq 1 - \varepsilon_i, \qquad \varepsilon_i \geq 0$$

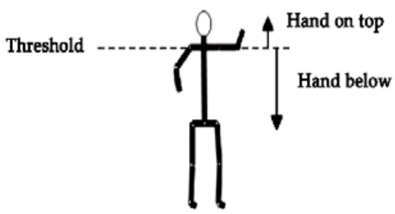

Figure 8. Threshold to Determine Human Pose [18]

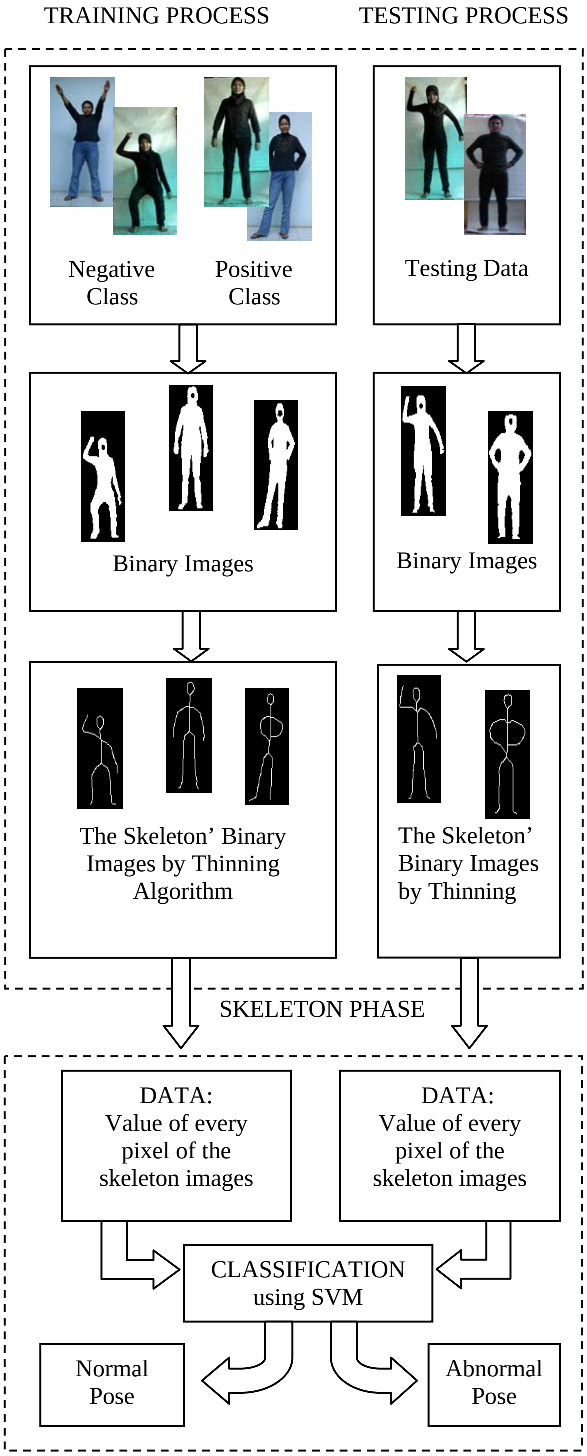

Figure 9. Human Pose Interpretation Using SVM

Here, training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function.

Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) \equiv \phi(x_i)^T(x_j)$ is called the kernel function.

Though new kernels are being proposed by researchers, four basic kernels in SVM method as follow:

1) Linear $K(x_i, x_j) = x_i^T x_j$.
2) Polynomial $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$
3) Radial Basis Function (RBF)
$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2, \gamma > 0.$
4) Sigmoid $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r).$

Here, γ , r, and d are kernel parameters [21]. In this paper, linear function is used as the kernel function.

3. Results and Discussion

The input data in the experiment are real images of human pose showing human behavior in front of cashier of convenient store. Currently, experiment is done in each sub system of HPIS. First experiment is human tracking sub system. In this experiment, two data videos are used. In each data video shows different sequence poses. The first data video consists of 312 frames and the second data consists of 468 frames. Example of these data is depicted in Figure 10.

In this first experiment, ordinate of human's location is assumed having constant value. Hence only abscissa of human's location is tracked using modified PF. In experiment, computational time of conventional PF and PF with binary Gaussian weighting is compared. Adaboost classifier [3] is used in the weight calculation in this experiment.

Since particles are generated randomly based on transition model, hence each experiment is done three times. Average computational time and tracking accuracy of conventional particle filter and particle filter with binary Gaussian weighting can be seen in Table 1 and Table 2.

Table 1 shows that PF with binary Gaussian weighting reduces computational time of conventional particle filter up to 68.34% in average. Since in the conventional method, more computational time is required to calculate weight in each predicted particle. Meanwhile, in the proposed method, time is only required to estimate mean and variance value to build Gaussian model, and weight is calculated based on the model.

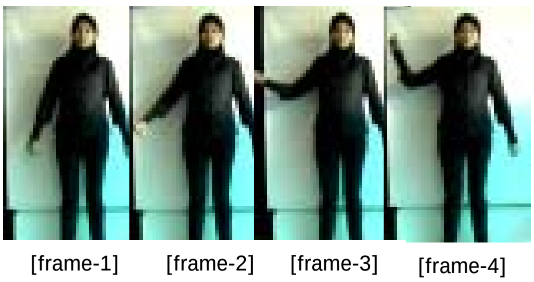

Figure 10. Example of Data Video for Human Tracking

Table 2 shows that tracking accuracy of proposed method is comparable with the conventional one. The conventional method has 93.89% and the proposed method has 90.03% of tracking accuracy. Hence, the proposed method could reduce the computational time of tracking and still have high tracking accuracy.

Second experiment is skeletonization and classification sub system. There are two poses in training and testing data, i.e. standing and sitting. Besides, it consists of two classes, positive class and negative class. Positive class is normal behavior when human is standing in front of cashier of convenient store. Conversely, negative class is abnormal behavior. Examples of positive class and negative class images are shown in Fig. 11.

Table 1. Computational Time Comparison in Three Experiments

Data Video	Experiment	t_{PF} (second)	t_{GPF} (second)	Time Reduction (%)
1	1	0.4308	0.1918	55.48
	2	0.4422	0.1902	56.99
	3	0.4136	0.1825	55.88
2	1	0.4169	0.1320	68.34
	2	0.4191	0.1426	65.97
	3	0.4232	0.1378	67.44
Average of time Reduction				61.68

Table 2. Tracking Accuracy Comparison in Three Experiments

Data Video	Experiment	Accuracy of PF (%)	Accuracy of GPF (%)
1	1	91.67	100.00
	2	92.95	100.00
	3	92.63	100.00
2	1	95.73	71.37
	2	96.15	84.19
	3	94.23	84.62
Average of Tracking Accuracy		93.89	90.03

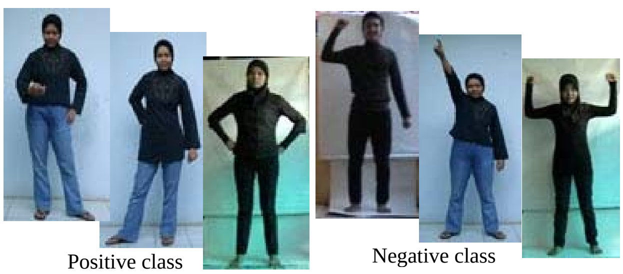

Figure 11. Example of Positive and Negative Class Images

Table 3. Experimental Result

Experiment	Σ Training data Positive Class	Σ Training data Negative Class	Σ Testing data Positive Class	Σ Testing data Negative Class	Σ Misclassification data	Degree of Accuracy (%)
1	13	31	14	30	10	77.27
2	14	30	13	31	8	81.82
3	14	30	13	31	7	84.09
Average of Accuracy						81.06

Result of this experiment is shown in Table 3. Table 3 shows the result of the three experiments. It shows that the SVM accuracy for classifying 44 testing data using 44 training data is 81.06%.

Based on the experiments, the number of misclassification data is 7 until 10 data. Misclassification data in positive class are more than in negative class. It happens because the training data in positive class is less than in negative class.

It is observed that misclassification images are positive class images which the hand of standing pose is the same as the hand of sitting poses. In fact, sitting pose includes negative class images. Moreover, misclassification images are also images which the pose is different from the images in training data. Because of the high similarity among the images, SVM can't exactly classify all of them.

4. Conclusion

Tracking experiment shows that the proposed method reduces computational time of particle filter up to 61.68% in average and the accuracy of tracking of the proposed method is comparable with conventional one, i.e. 90.03%. This reduction time is achieved since in the proposed method, time is required only to estimate mean and variance value to build Gaussian model and weight is calculated based on the model. Meanwhile in the conventional method, time is required to calculate particle's weight one by one.

Through the skeletonization and classification experiment using 44 real human pose images, the subsystem achieves 81.06% of classification accuracy. However, it can be concluded that the thinning algorithm is simple but not too effective because if image data is not same as the assumed image, it should be modified.

Due to the increasing usage of CCTV camera for security surveillance, combination of particle filter with binary Gaussian weighting for tracking and SVM for the interpretation can be further developed for an automatic crime detection system.

References

[1] Criminal data, *http://www.tempointeraktif.com/hg/nasional/2005/01/01/brk,20050101-01,id.html*, accessed on 2 November 2007.

[2] Oblligation of using CCTV, *http://www.ispy.co.id/mos/index.php?option=com_content&task=view&id= 43&Itemid=1*, accessed on 15th April 2008.

[3] I. Laptev, Object Detection with Boosted Histograms, IRISA/INRIA, Rennes, France, January 19, 2007.

[4] C.Y. Suen, T.Y. Zhang, Communications of the ACM, 27/3 (1984) 236.

[5] C. Fatihah, Tesis, Fakultas Ilmu Komputer, Universitas Indonesia, Indonesia, 2007.

[6] M.S. Arumpalam, S. Maskell, N. Gordon, T. Clapp, IEEE Transaction on Signal Processing, 40/2 (2002) 174.

[7] Y. Iwahori, T. Takai, H. Kawanaka, H. Itoh, Y. Adachi, Particle Filter Based Tracking of Moving Object from Image Sequence, KES 2006, Part II, LNAI 4252, Springer-Verlag, Berlin Heidelberg, 2006, p.401.

[8] E.B. Meierm, F. Ade, Tracking Cars in Range Images Using Condensation Algorithm, Technical Report in Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.

[9] C.K. Ho, Short Introduction to Particle Filter, *www.sps.ele.tue.nl/members/C.K.Ho/*, accessed on February 2007.

[10] K. Kawamoto, K. Hirota, N. Wakami, Efficient and Robust Curve Tracker based on Particle Filtering in Digital Space, ISIS, Korea, September 2005.

[11] P. Fearnhead, Ph.D Thesis, University of Oxford, 1998.

[12] N.J Gordon, D.J Salmon, A.F.M Smith, IEE Proceedings-F, 140/2 (1993) p.107.

[13] K. Kawamoto, International Symposium on Advanced Intelligent System, Korea, September 2005.

[14] I. Agustien, M.R. Widyanto, International Conference on Soft Computing, Intelligent System and Information Technology, Bali, Indonesia 2007.

[15] I. Agustien, M.R. Widyanto, 4th International Conference Humanoid, Nanotechnology,

Information Technology Communication and Control, Environment and Management (HNICEM), Manila, Philipines, March 12-19, 2009.
[16] C.K. Ho, Introduction to Particle Filter, Lecture Notes of High Tech Campus Eindhoven, Holland, 2005.
[17] W.R. Leo, Statistics and the Treatment of Experimental Data, Adapted from Chapter 4, Technique for Nuclear and Particle Physics Experiments, Springer-Verlag, Berlin Heidelberg, 1992, p.62.
[18] M.R. Widyanto, S.N. Endah, K. Hirota, 4th International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM), Manila, Philipines, March 12-19, 2009.
[19] B.E. Boser, I. Guyon, V. Vapnik, Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania, United States, ACM Press, 1992, p.144.
[20] C. Cortes, V. Vapnik, Support-vector Network, Machine Learning 20 (1995) 273.
[21] C. Hsu, C.C. Chang, C. Lin, A Practical Guide to Support Vector Classification, Department of Computer Science, National Taiwan University, Taiwan, 2008.