

An Improved Mean Shift Using Adaptive Fuzzy Gaussian Kernel for Indonesia Vehicle License Plate Tracking

Basuki Rahmat, *Member, IAENG*, Endra Joelianto, I Ketut Eddy Purnama, and Mauridhi Hery Purnomo, *Member, IAENG*

Abstract—A new approach toward Indonesian vehicles license plate tracking based on video recordings of vehicles on the highway, is proposed. The tracking technique is used to improve the performance of a standard Mean Shift with a Gaussian kernel by selecting the appropriate kernel radius using an adaptive fuzzy mechanism. The purpose of kernel radius variation of Parzen window is to keep or maximize the mean of the similarity function outputs which implies a successful tracking process. The experimental results show that Improved Mean Shift using Adaptive Fuzzy Gaussian Kernel proved to have better effects as compared to the Standard Mean Shift.

Index Terms—Improved, Mean Shift, Adaptive, Fuzzy, Gaussian, Kernel radius, Parzen window

I. INTRODUCTION

OBJECT tracking is one of the most important components that can be applied in the field of computer vision-based video. The tracking-based video is the task of estimating time position of the object analyzed in a sequence of images. Object tracking has always been exciting and challenging for researchers who want to analyze video-based objects. Video-based object tracking research continues to be developed, among others [1]–[7].

Video tracking is pursuing an object or objects in a series of multi-sequence of video frames. An essential problem that is often faced in computer vision applications is object tracking. This is due to the difficulties in tracking objects that can arise from internal and external factors such as disfigurement, motion cameras, motion-blur, and occlusion [8]. The major challenge that must be considered when designing an object tracking system is the appearance of the object model and the candidate model in the scene recorded on video that is often similar, and including the variations in

appearance of the object itself.

In general, several methods of vehicle license plate tracking or object tracking can be categorized based on the features that are used, among other things:

- a. Color Feature [9], [10].
- b. Superpixels Feature [11].
- c. Oriented FAST and Rotated BRIEF (ORB) Feature Matching [12].
- d. Edges Feature, Optical flow Feature, and Texture Feature [10].

Three major parts are developed in the process of vehicle license plate tracking and recognition system through video surveillance as shown in Fig.1, namely:

- License Plate Extraction part to get the vehicle license plate's location contained in the video frame.
- License Plate Tracking part to track the vehicle license plate location along multiple video frames, and
- Character Extraction part, in which there are problems of license plate character segmentation and recognition.

This paper includes the study on license plate tracking by using an Improved Mean Shift with Gaussian kernel which is governed by an adaptive fuzzy mechanism.

The main problem associated with license plate tracking video data is the possibility of objects other than the license plate itself appearing as background image that might be similar to the target appearance which could cause interference with observations. In such cases, it may be difficult to distinguish the features of the expected target, which produces a phenomenon known as clutter. In addition to the challenges due to clutter, it is also worth noting some of these following factors [13]:

- a. Posing Variation. Non-stationary target will have different variations when it is projected onto the image plane, such as when turning or changing direction.
- b. Ambient illumination. Direction, intensity, and color of the light from surrounding environment affect the appearance of the target. Also, global illumination changes often give problems for video recording data in the outdoors. For example, the light environment will have a different effect when the light from the sun is obscured by clouds. In addition, the angle between the light direction and the normal to the surface of the object pose different objects, which will affect vision through the camera lens.
- c. Noise. The acquisition process consists of a series of images in the video frames may introduce noise, depending on the quality of the camera used to capture.

Manuscript received May 7th, 2017; revised July 28th, 2018. This work was supported by Directorate General of Higher Education, Ministry of Research and Education, Indonesia (doctoral fellowship program).

Basuki Rahmat is a student doctoral program of Electrical Engineering Department, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. He works as a lecturer of Informatics Engineering Departement of Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia. (email: basukirahmat.if@upnjatim.ac.id).

Endra Joelianto is with the Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Bandung 40132, Indonesia (e-mail: ejoel@tf.itb.ac.id).

I Ketut Eddy Purnama and Mauridhi Hery Purnomo are with the Electrical Engineering Department, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, 60111 (e-mail: {ketut, hery@ee.its.ac.id}).

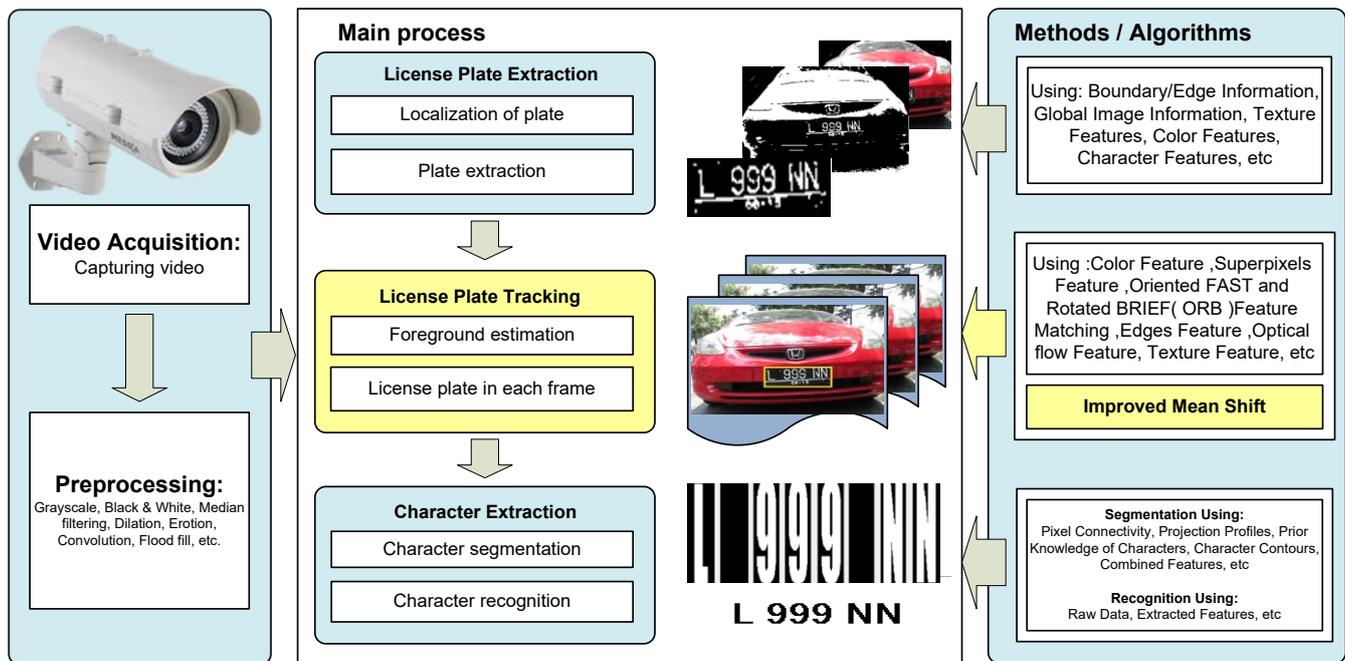


Fig. 1. Video-based vehicle license plate tracking and recognition system.

Object observation due to noise could have disrupted some of the data which degraded system performance.

- d. Occlusions. Observation failure could also happen when the object is partially or completely clogged (blocked) by other objects. Failure to observe the target due to obscuration or clogging by other objects that may be present in the scene.

Also, there are other important issues related to the license plate itself. Things to note from the problems of the vehicle license plate or the environment are the detection and the vehicle license plate recognition. The problems are described as follows [14]:

1). Vehicle license plate variations

- a) location (position): a plate of the vehicle is in a different position for each vehicle;
- b) quantity: a picture of the vehicle which might have more than one plate;
- c) size: vehicle license plates could have a variety of sizes depending on the distance captured by the lenses. It is also important to take into account zoom factor;
- d) color: vehicle license plates could have many different characters (letters and numbers) and background colors according to the type of plate or imaging equipment;
- e) letters: vehicle license plates' letters could vary in one country and another due to different usage of fonts and languages;
- f) standard and nonstandard: each country has its own standard rules of license plate numbering, however, many found a lot of vehicles are using non-standardized license plates;
- g) occlusion: vehicle number plates may be shrouded by dirt;
- h) trends: vehicle license plates can be tilted;

- i) others: the addition of characters, vehicle license plates, and screws in frames.

2). Environmental variations

- a) illumination: input image may have a variation of lighting due to environments and vehicle lights;
- b) background: the background image may have a pattern similar to the vehicle license plate, such as the number on the vehicle, the bumper with a vertical pattern, and formed grounds.

One algorithm in intelligent systems that is constantly being developed to date and likely developed steadily for the foreseeable future to solve problems especially for video-based object tracking is Mean Shift algorithm. Some examples of the application of this algorithm to solve the problems of video-based object tracking, among others, can be found in the following papers [1],[5],[15]–[17].

The Mean Shift algorithm serves as an iterative algorithm, and it is powerful and versatile. It has a nonparametric nature that allows easier combinations and integrations with other algorithms. Some previous results have managed to improve the performance of Mean Shift by adding technical or other algorithms, among others can be found in [9], [12], [18]–[26].

This paper serves as a proposal to achieve better results of the standard Mean Shift with the Gaussian kernel by selecting appropriate kernel radius using adaptive fuzzy mechanism. Some of the reasons for using adaptive fuzzy include flexible systems, capable of modeling complex nonlinear functions, and working with other techniques [27]–[31]. These Mean Shift and adaptive fuzzy combination algorithms are to solve the problem of video-based Indonesian vehicle license plate tracking.

Improved Mean Shift performance can be accomplished by using adaptive fuzzy system mechanism based on the Probability Density Function (PDF) of the object model and

PDF of a candidate model. The proposed method is based on the consideration that the radius of Parzen window kernel is one of the important and influential parts in the generating of the PDF of the object model and the candidate model. By changing the radius, the proposed algorithm attempts to maintain the highest performance in tracking an object in terms of similarity evaluation described by the accuracy of tracking called percentage accuracy of object tracking (PAOT).

II. PRELIMINARIES

A. Probability Density Function (PDF)

The mathematical definition of the continuous probability function, $p(s)$, satisfies the following properties [32]:

1. The probability that s is between two points a and b .

$$p(a < s < b) = \int_a^b p(s) ds \quad (1)$$

2. Values are not negative for all real s .
3. The integral of the probability function is one:

$$\int_{-\infty}^{\infty} p(s) ds = 1 \quad (2)$$

The most commonly used probability function is the Gaussian function (also known as the normal distribution):

$$p(s) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s-c)^2}{2\sigma^2}\right) \quad (3)$$

Extending to the case of vector s , owned by non-negative $p(s)$ with the following properties:

1. The probability that s is within \mathfrak{R} region is:

$$P = \int_{\mathfrak{R}} p(s) ds \quad (4)$$

2. The integral of the probability function is one, that is:

$$\int p(s) ds = 1 \quad (5)$$

With a set of n samples of data s_1, \dots, s_n , We can use a method called density estimation in which the density function $p(s)$ can be estimated, then the output of $p(s)$ for each upcoming sample s is obtained. The basic idea behind many methods to estimate the probability of an unknown density function is very simple. This technique depends on the probability P that a vector falls in a region \mathfrak{R} as given by Eq.(4).

If it is assumed that \mathfrak{R} is so small that $p(s)$ is not much different in it, then it can be written:

$$P = \int_{\mathfrak{R}} p(s) ds \approx p(s) \int_{\mathfrak{R}} ds = p(s)V \quad (6)$$

Where V is the volume of \mathfrak{R} .

Suppose n samples s_1, \dots, s_n are pulled independently according to the probability function of density $p(s)$, and there k of n samples within the region \mathfrak{R} , then the relationship is obtained:

$$P = k/n \quad (7)$$

So come to the following obvious estimates for $p(s)$:

$$p(s) = \frac{k/n}{V} \quad (8)$$

B. Kernel Density Estimation

Kernel Density Estimation (KDE) is a method to estimate PDFs of random variables through non-parametric way. The kernel density estimation is a matter of basic data smoothing where population conclusions are made, based on a limited sample of data. Areas such as signal processing and econometrics, this method is also called the Parzen-Rosenblatt window method, named after Emanuel Parzen and Murray Rosenblatt, which is credited with making it independent in its current form [33], [34].

It is further considered that \mathfrak{R} is an s centered hypercube (2-D square), as shown in Fig.2. The length of the edge of the hypercube can be represented by h , in which $V = h^2$ for the 2-D square, and $V = h^3$ for a 3-D cube [32].

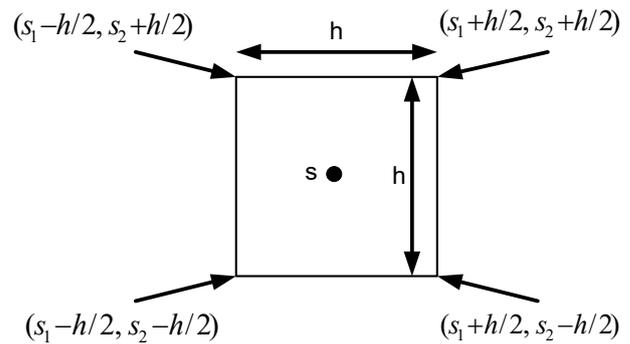


Fig. 2. Hypercube square 2-D.

An equation was introduced [32]:

$$g\left(\frac{s-s_i}{h}\right) = \begin{cases} 1 & \frac{|s_k - s_{ik}|}{h} \leq 1/2, k=1,2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Indicates whether s_i is in the box (centered on s , with width h) or not. The number of samples k falling within the region \mathfrak{R} , from n , is given by:

$$k = \sum_{i=1}^n g\left(\frac{s-s_i}{h}\right) \quad (10)$$

So by using Eq.(8), and $V = h^2$ for a 2-D square, the Parzen probability density estimation formula (for 2-D) is obtained:

$$p(s) = \frac{k/n}{V} = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^2} g\left(\frac{s-s_i}{h}\right) \quad (11)$$

$g\left(\frac{s-s_i}{h}\right)$ is called window function [32].

Then for the 3-D cube hypercube, it is assumed that the area \mathfrak{R} which encloses the k sample from n , is the hypercube with the long side h centered on s , as shown in Fig.3, with volume $V = h^3$. Then the total number of points inside the hypercube, as in Eq.(10). Using Eq.(8), and $V = h^3$ for 3-D cube, the Parzen probability density estimation formula (for 3-D) is obtained:

$$\begin{aligned}
 p(s) &= \frac{k/n}{V} \\
 &= \frac{1}{n} \sum_{i=1}^n \frac{1}{h^3} g\left(\frac{s-s_i}{h}\right)
 \end{aligned}
 \tag{12}$$

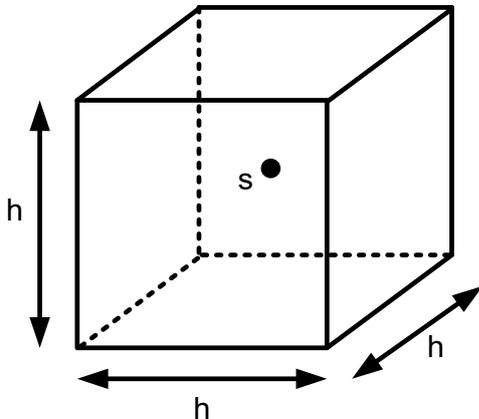


Fig. 3. Hypercube cube 3-D.

Thus in general with dimension D, the Parzen probability density estimation formula is obtained [35]:

$$p(s) = \frac{1}{n h^D} \sum_{i=1}^n g\left(\frac{s-s_i}{h}\right)
 \tag{13}$$

Previously mentioned, that $g((s-s_i)/h)$ is called window function. In practice, $g(\bullet)$ in this window function, usually the kernel function is used. This kernel, which corresponds to a center-based hypercube unit, is known as the Parzen window or naïve estimator [35]. The quantity $g((s-s_i)/h)$ equals the unity if s_i is in the hypercube with the side h centered on s , and equal to zero if it is outside the hypercube. With $g(\bullet)$ in the window function using the kernel function, the Parzen probability density estimate with dimension D in Eq.(13) is also called Kernel Density Estimation (KDE), expressed as in Eq.(14) [35].

$$\begin{aligned}
 p_{KDE} &= p(s) \\
 &= \frac{1}{n h^D} \sum_{i=1}^n g\left(\frac{s-s_i}{h}\right)
 \end{aligned}
 \tag{14}$$

The h side will generally be if the window function uses kernel function, it is called kernel bandwidth [36], [37] or kernel radius [9].

C. Kernel Function

Estimation of PDF by using KDE always involves the kernel function. The kernel function $g(\bullet)$ satisfies the properties of the continuous probability function, i.e. the real nonnegative value, and the integral of the probability function is one:

$$\int_{-\infty}^{\infty} g(u) du = 1
 \tag{15}$$

Some commonly used kernel functions include:

1. Uniform

$$g(u) = \begin{cases} 1/2, & \text{for } |u_i| \leq 1 \\ 0, & \text{for } |u_i| \text{ others} \end{cases}
 \tag{16}$$

2. Triangular

$$g(u) = \begin{cases} 1-|u|, & \text{for } |u_i| \leq 1 \\ 0, & \text{for } |u_i| \text{ others} \end{cases}
 \tag{17}$$

3. Epanechnikov

$$g(u) = \begin{cases} 3/4(1-u^2), & \text{for } |u_i| \leq 1 \\ 0, & \text{for } |u_i| \text{ others} \end{cases}
 \tag{18}$$

4. Gaussian

$$g(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right), \text{ for } |u_i| < \infty
 \tag{19}$$

D. Kernel Radius or Bandwidth Selection

As mentioned previously, from Eq.(14) there is a parameter h in the PDF estimation by using KDE. This h parameter is very important and influences the success of PDF estimation. The h parameter of this kernel function is called kernel bandwidth [36], [37] or sometimes also called kernel radius [9].

To illustrate the effect of the parameter h on the success of the PDF estimation, a random sample of standard normal distributions as given in Eq.(3) is can be seen on the x -axis (horizontal), as shown in Fig.4. The gray curve is the actual density (normal density average 0 and variant 1). When compared, the red curve looks worse because there are a lot of artificial data artifacts that is coming from the use of kernel radius $h = 0.05$, which is too small. The over smooth green curve because using the kernel radius $h = 2$ shrouds many of the foundation structures. The black curve with the kernel radius of $h = 0.337$ is optimally refined because the approximate density is close to the actual density.

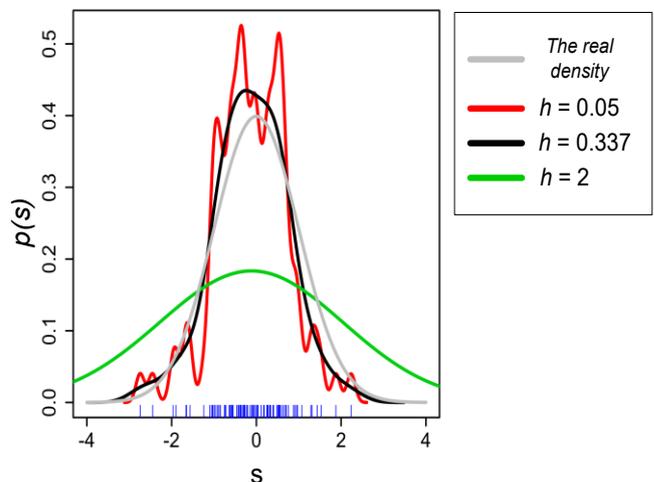


Fig. 4. Influence of kernel radius change.

If the kernel radius h is expressed as a percent, a 3-D visualization, the kernel radius' effect on the mask of a Parzen window using Gaussian kernel, with the respective radius, $h = 65\%$, 75% , 85% and 95% as shown in Fig. 5.

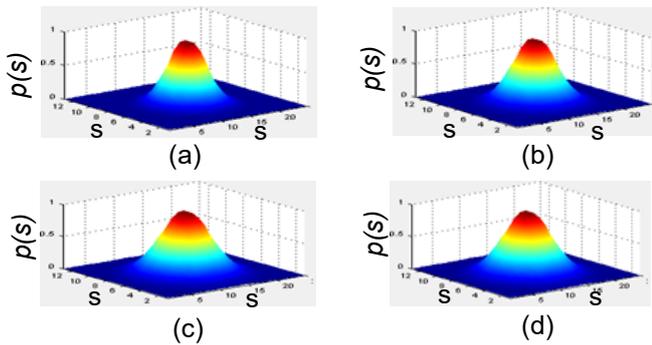


Fig. 5. The shape of the mask of a Parzen window using Gaussian kernel, with the respective radius, $h = 65\%$, 75% , 85% and 95% .

E. Mean Shift Basic

Mean Shift, proposed by Fukunaga and Hostetler in 1975, is a non-parametric estimation strategy [38]. It can be extended to locate the ideal match between the object model and the candidate model in light of the gradient estimation of the PDF estimation in the kernel [39]. The similarity function is used to assess the similarity of PDF of the object model and the candidate model.

To understand how the Mean Shift works, it is assumed intuitively as to the identical distribution of billiard balls as shown in Fig.6 [40]. The goal is to find the densest or the highest distributed billiard distribution density. For example, an arbitrary location is chosen, i.e. the region of interest (ROI) as shown in Fig.6. From the position at the center point of the ROI, the center of mass of the spherical distribution is known. Furthermore, from the center point of ROI shifts towards the center of the mass. Leaving the Mean Shift vector trace. The previous mass distribution center of the ball is now the center of ROI. From this position at the new center of ROI, the mass center of the ball distribution is known again. Furthermore, the new center of ROI was shifted again towards the new center of mass which leaves the Mean Shift vector trace again. And so on until the center point of ROI is equal to the mass center of sphere distribution, so there is no more shift. This means that the highest density of the highest distribution of billiard balls has been found.

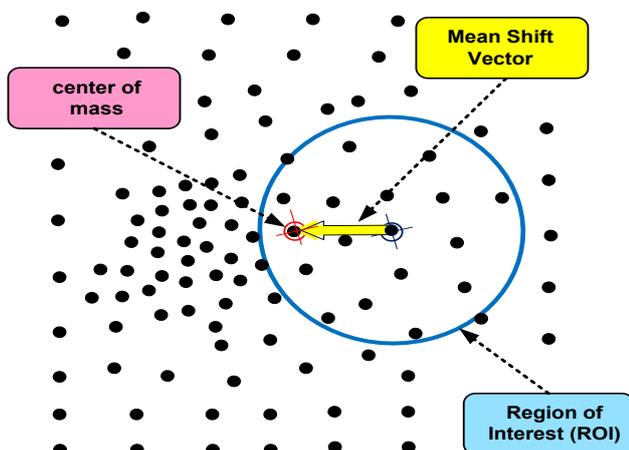


Fig. 6. The distribution of identical billiard balls.

Shifting means considering feature space as an empirical PDF. Assuming that the input for the Mean Shift is a set of points, it then can be thought as a sample of the underlying PDF. If a solid area (or cluster) exists in the feature space, then it corresponds to the (or maxima local) mode of the PDF. Groups associated can also be identified through the association with the given mode using Mean Shift.

Considering each data point, it is important for the Mean Shift to associate the nearest peak of the PDF dataset. Meanwhile, Mean Shift will then analyze the data points' average. Then, it moves the center of the window to the average. Repeat until convergent. At each iteration, it may be considered that the window shifts to a denser area of the dataset.

Thus, Mean Shift works as follows:

1. Fix the window around each data point.
2. Calculated average data in the window.
3. Slide the window toward the average.
4. 1-3 repeated until convergent.

F. Mean Shift Tracking

For object tracking problems, color features based Mean Shift technique have been widely used. The PDF as the color probability functions is obtained by using Parzen window using conversion from RGB to indexed colors.

The probability of the color feature u of the object model q_u , and the probability of the color feature u from the candidate model $p_u(s)$, can be expressed as in Eq.(20) and Eq.(21) [40].

$$q_u = C \sum_{b(s_i)=u} g(\|s_i\|^2) \quad (20)$$

$$p_u(s) = C_h \sum_{b(s_i)=u} g\left(\left\|\frac{s-s_i}{h}\right\|^2\right) \quad (21)$$

where

- u : color feature.
- s : the object's center spot in the previous video frame.
- s_i : the pixel location within the kernel in the current frame.
- $g(\bullet)$: kernel function.
- C : normalization factor of the object model.
- C_h : normalization factor of the candidate model.
- $b(s_i)$: the color index (1..m) of s_i .
- h : kernel radius.
- q_u : the probability of the color feature u of the object model.
- $p_u(s)$: the probability of the color feature u from the candidate model.

The probability of the color feature u of the object model q_u , and the probability of the color feature u of the candidate model $p_u(s)$, is sometimes also expressed by the Kronecker delta function as in the following equation [9]:

$$q_u = C \sum_{b(s_i)=u} g(\|s_i\|^2) \delta[b(s_i)-u] \quad (22)$$

$$p_u(s) = C_h \sum_{b(s_i)=u} g\left(\left\|\frac{s-s_i}{h}\right\|^2\right) \delta[b(s_i)-u] \quad (23)$$

Where \mathcal{D} is the Kronecker delta function.

The relationship between the two PDFs which is the probability of the color feature can be expressed by the similarity function of $f_1(s)$ or Bhattacharyya coefficient $f_2(s)$ as in Eq.(24) [9] and Eq.(25) [36]:

$$f_1(s) = \sum_{u=1}^m \sqrt{q_u / p_u(s)} \quad (24)$$

$$f_2(s) = \sum_{u=1}^m \sqrt{p_u(s) q_u} \quad (25)$$

The candidate models would be shifted towards Mean Shift in a repeatable way to maximize the similarity parameter. Mean Shift has local convergent iterations rise toward higher density of a given probability distribution. The iteration process repeatedly goes until the highest density of the optimal estimation of the object location is achieved which indicates the object has been tracked successfully.

The computation of Mean Shift vector to translate the Kernel window by $m(s)$, is expressed as Eq.(26) [41].

$$m(s) = \left[\frac{\sum_{i=1}^n s_i g\left(\frac{\|s-s_i\|^2}{h}\right)}{\sum_{i=1}^n g\left(\frac{\|s-s_i\|^2}{h}\right)} - s \right] \quad (26)$$

where

- $m(s)$: the desired Mean Shift vector in one iteration within the kernel.
- s : the object's center spot in the previous video frame.
- s_i : The current frame's pixel spot inside the kernel.
- $g(\bullet)$: kernel function.
- h : kernel radius.

Next, to get the Mean Shift gradient is based on the slope of the density contour gradient. The generic formula for gradient slope as in Eq.(27) [42]:

$$s_1 = s_0 + \eta f'(s_0) \quad (27)$$

If applied to the KDE equation as in Eq.(14) then the relationship will be obtained as follows:

$$p_{KDE} = \frac{1}{n h^D} \sum_{i=1}^n g\left(\frac{s-s_i}{h}\right)$$

$$\nabla p_{KDE} = \frac{1}{n h^D} \sum_{i=1}^n g'\left(\frac{s-s_i}{h}\right)$$

Setting it to 0, we get [42]:

$$\sum_{i=1}^n g'\left(\frac{s-s_i}{h}\right) \vec{s} = \sum_{i=1}^n g'\left(\frac{s-s_i}{h}\right) \vec{s}_i$$

Thus, the Mean Shift gradient as follows [42]:

$$\vec{s} = \frac{\sum_{i=1}^n g'\left(\frac{s-s_i}{h}\right) \vec{s}_i}{\sum_{i=1}^n g'\left(\frac{s-s_i}{h}\right)} \quad (28)$$

Furthermore, to obtain the Parzen window mask and its gradient against the x-axis and y-axis, with various kernel functions used: Uniform, Triangular, Epanechnikov, and Gaussian, using the Parzen_window function as in Eq.(29) [43].

$$[km, gx, gy] = \text{Parzen_window}(tm, lm, h, g(\bullet), \text{graph}) \quad (29)$$

Where

- km : Parzen window mask.
- gx, gy : Parzen window gradient against the x and y-axes.
- tm, lm : height and width of Parzen window mask size.
- rm : Parzen window mask radius.
- $g(\bullet)$: kernel function.
- graph : plot the image mask if $\text{graph} = 1$.

Finally to get the Mean Shift tracking is by using the MeanShift_tracking function as in Eq.(30) [43].

$$[s_i, \text{loss}, f, f_indx] = \text{MeanShift_tracking}(q_u, I2, \dots, Lmap, \text{height}, \text{width}, f_thresh, \text{max_it}, s, \dots, tm, lm, km, gx, gy, f, f_indx, \text{loss}) \quad (30)$$

where

- s : the center location of the object in the previous video frame.
- s_i : the pixel location within the kernel in the current frame.
- loss : the object out of tracking
- (f, f_indx) : the storage of the result of similarity function during the tracking process.
- q_u : the probability of the color feature u of the object model.
- $I2$: next frame.
- $\text{height}, \text{width}$: size of $I2$.
- $Lmap$: colormap length.
- f_thresh : the threshold value of the similarity function.
- max_it : maximum iteration.
- tm, lm : height and width of Parzen window mask size.
- km : Parzen window mask.
- gx, gy : Parzen window gradient against the x and y-axes.

III. PROPOSED METHODOLOGY

The success of the object tracking process in general or the license plate tracking process, in this case, is marked by the success of maintaining the value of similarity functions throughout the video frame. In this paper, we propose a new method to keep the value of the desired similarity function by selecting appropriate kernel radius during the tracking process along the video frame. The selection process of the kernel radius follows the change in the value of the similarity

function. The technique for obtaining the appropriate kernel radius is by using the adaptive fuzzy system adjustment mechanism. So far the use of kernel radius is static during the tracking process along the video frame. So the success of the tracking process is not consistent depending on the selection of kernel radius.

This paper contributes mainly the selection of appropriate and dynamic kernel radius through an adaptive fuzzy mechanism based on the reading of similarity function values. The result of the tracking process is expected to be better than the use of static kernel radius. Since the kernel radius selection uses an adaptive fuzzy mechanism and the kernel function used is Gaussian, this method is then called Adaptive Fuzzy Gaussian kernel.

Furthermore, to assess the similarity function, it should be compared to an arbitrary value, e.g., \mathcal{E} . Value (\mathcal{E}) is chosen and compared to the mean of the output similarity function in one frame of video. The value of \mathcal{E} is obtained from the mean of similarity function output from the standard Mean Shift tracking process using Gaussian kernel function. The changes of the kernel radius are needed to keep the value of the similarity fixed or improved compared with a certain value, \mathcal{E} . Increasing or decreasing the value of the similarity to the value of \mathcal{E} is the basis for increasing or decreasing the kernel radius.

By Eq.(24), the value of \mathcal{E} is obtained from standard Mean Shift tracking process by using Gaussian kernel function in the same video. With Z as a number of the frame, then the value of \mathcal{E} is defined as Eq.(31).

$$\mathcal{E} = \frac{\sum_{i=1}^Z f_1(s)_i}{Z} \quad (31)$$

The mean value of similarity function output is compared with the value of \mathcal{E} . If the mean value of similarity function output is the same or greater than the value of \mathcal{E} , then the tracking process is continued. Conversely, if it is smaller than the value of \mathcal{E} , the value of difference will then be obtained. The value of the difference is then used as an error and a delta error values. The error and delta error values are used as a basis for the use of the adaptive fuzzy system. With the mechanism of fuzzy rules, then the new kernel radius value will be obtained. The process is then continued using the new kernel radius value. The process is repeated until the maximum iteration has been reached.

A. Algorithm Steps

The algorithm process explained are:

- Step 1:** Initialization of the Mean Shift tracking algorithm. Once License plate object area is selected, then the probability distribution of histogram is determined for the object model.
- Step 2:** The candidate model as the estimation is determined for the current frame.
- Step 3:** Mean-Shift tracking process. The similarity function between PDF of the object model and the candidate model is calculated. The mean value of similarity function output is obtained.

- Step 4:** The mean value of similarity function output is compared with the value of \mathcal{E} . If the mean value of similarity function output is the same or greater than the value of \mathcal{E} , then the tracking process is continued. Conversely, if it is smaller than the value of \mathcal{E} , then the difference value is obtained. The difference value will be used as the error and the delta error, and performing adaptive fuzzy system process. By using the adaptive fuzzy mechanism, then the new kernel radius value will be obtained. After that, the process continued using the new kernel radius value.
- Step 5:** The process is repeated until the maximum iteration has been reached.

The flow diagram of the proposed method is shown in Fig. 7.

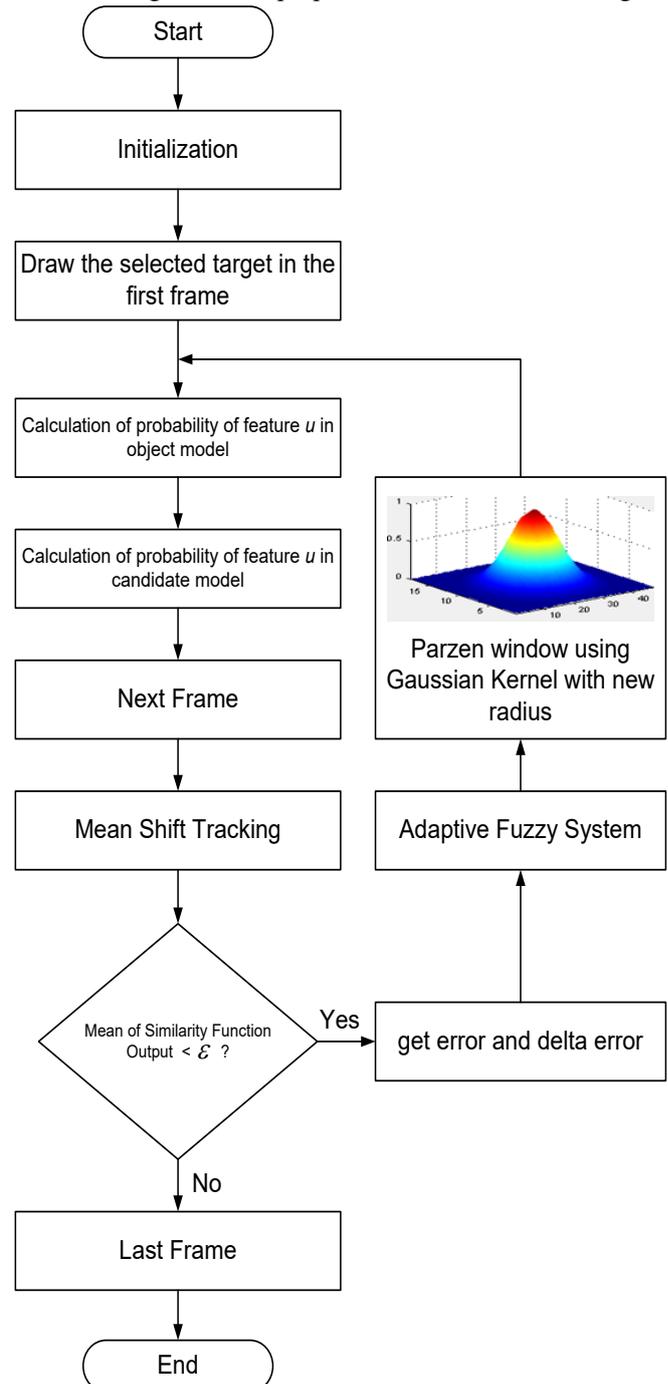


Fig. 7. Flow chart proposed methodology.

B. Adaptive Fuzzy Design

The design of the adaptive fuzzy system for the improved Mean Shift using adaptive fuzzy Gaussian kernel for video-based Indonesia vehicle license plates tracking is shown in Fig. 8.

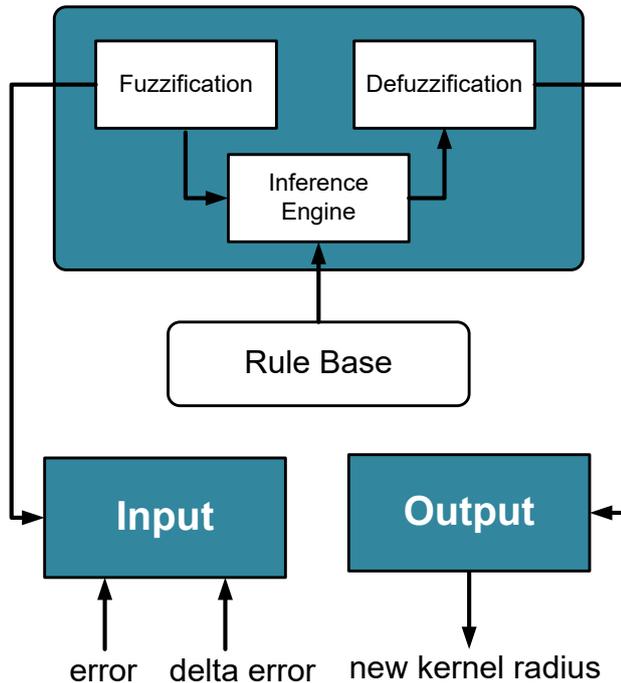


Fig. 8. Adaptive Fuzzy System.

These terms represent: small error (SE), medium error (ME), big error (BE), small delta error (SDE), medium delta error (MDE) and big delta error (BDE), and the Gaussian function used, is designed as the membership function. The Gaussian membership functions are defined by Eq.(32).

$$\begin{aligned}
 \mu_{SE} &= \text{gaussian}(\text{error}; \sigma_1, c_1) \\
 \mu_{ME} &= \text{gaussian}(\text{error}; \sigma_2, c_2) \\
 \mu_{BE} &= \text{gaussian}(\text{error}; \sigma_3, c_3) \\
 \mu_{SDE} &= \text{gaussian}(\text{delta error}; \sigma_4, c_4) \\
 \mu_{MDE} &= \text{gaussian}(\text{delta error}; \sigma_5, c_5) \\
 \mu_{BDE} &= \text{gaussian}(\text{delta error}; \sigma_6, c_6)
 \end{aligned} \quad (32)$$

The formulation of Gaussian with the membership function parameters σ_i, c_i and x as error or delta error is given by Eq. (33).

$$\text{gaussian}(x; \sigma_i, c_i) = \exp\left(-\frac{(x-c_i)^2}{2\sigma_i^2}\right), \text{ where } i=1..6 \quad (33)$$

Where the value of parameters σ_i, c_i respectively defined as Eq. (34).

$$\begin{aligned}
 \sigma_1 &= \max(\text{error}); & c_1 &= \max(\text{error})/3; \\
 \sigma_2 &= \max(\text{error}); & c_2 &= \max(\text{error})/2; \\
 \sigma_3 &= \max(\text{error}); & c_3 &= \max(\text{error})/1; \\
 \sigma_4 &= \max(\text{delta error}); & c_4 &= \max(\text{delta error})/3; \\
 \sigma_5 &= \max(\text{delta error}); & c_5 &= \max(\text{delta error})/2; \\
 \sigma_6 &= \max(\text{delta error}); & c_6 &= \max(\text{delta error})/1;
 \end{aligned} \quad (34)$$

Then the shape of the input fuzzy membership functions is shown in Fig. 9.

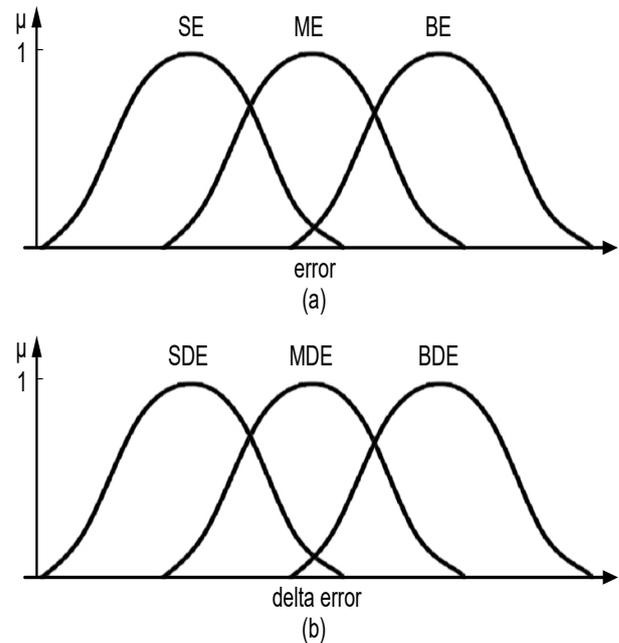


Fig. 9. Fuzzy membership functions of input error and delta error.

The use of fuzzy membership function parameter as in Eq.(34), is adaptive in accordance with error and delta error result of a difference of the mean of similarity function with a value of \mathcal{E} . So the Gaussian fuzzy membership function Eq.(32) is also adaptive. Changes in errors and delta errors which means changes to the fuzzy membership function result in a change in the new kernel radius. The goal is to return the ideal value of the mean of similarity function, under the value of \mathcal{E} (as in Fig. 7).

Furthermore, the shape of the output fuzzy membership function is designed as shown in Fig. 10.

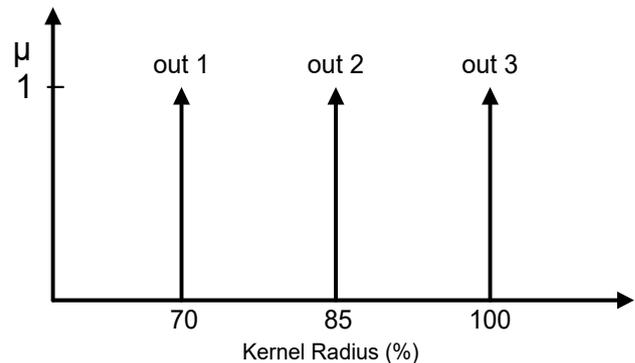


Fig. 10. Output fuzzy membership function.

According to Fig. 10, the crisp value of each out1, out2 and out3 is defined as the singleton fuzzy output in the form of a value choice of the kernel radius using Gaussian kernel (%), which is given as Eq. (35).

$$\text{out1} = 70; \text{ out2} = 85; \text{ out3} = 100; \quad (35)$$

The rule evaluation of adaptive fuzzy output is designed as shown in Table I.

TABLE I
RULE EVALUATION OF ADAPTIVE FUZZY OUTPUT

		Error		
		SE	ME	BE
Delta Error	SDE	out3	out2	out1
	MDE	out3	out2	out1
	BDE	out3	out2	out1

TABLE II
MEAN SHIFT PARAMETERS

Parameter	Value
the threshold for the similarity function	0.16
maximum number of iterations	100

Defuzzification calculation as fuzzy system output using Center Average Defuzzifier [44]. Suppose that the outputs are in the form of two fuzzy membership functions as in Fig.11. Where μ_1 and μ_2 are the Gaussian membership function value of the input error and delta error, out_1 and out_2 are corresponding fuzzy output values, and N is the number of Gaussian membership functions is processed. Then the output of the fuzzy system y^* using Center Average Defuzzifier can be calculated using Eq.(36) [44].

$$y^* = \frac{\sum_{i=1}^N out_i \mu_i}{\sum_{i=1}^N \mu_i} \tag{36}$$

In this example a defuzzification output is obtained:

$$y^* = \frac{\sum_{i=1}^2 out_i \mu_i}{\sum_{i=1}^2 \mu_i}$$

The illustration is shown in Fig.11.

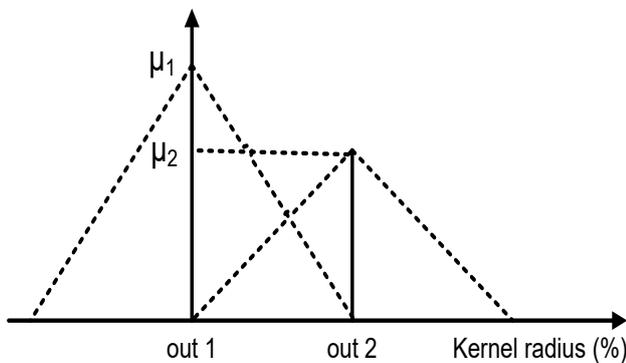


Fig. 11. Sample overview of center average defuzzifier.

IV. RESULTS AND DISCUSSION

Improved Mean Shift algorithm with Gaussian kernel function that has been added Fuzzy Adaptive mechanism tested on four video recordings of vehicles running on the highway. Vehicles used for testing are vehicles with license plate number of Indonesia. The algorithm is applied for tracking the license plate of the vehicle along the video frame. The parameters used for the test are presented in Table II.

The following figures give an overview of the simulation results. For Video 1, the following results are obtained, as shown in Fig. 12 through Fig. 15. The standard Mean Shift generating percentage accuracy of object tracking of 50.4673%, while the Mean Shift with adaptive fuzzy Gaussian kernel yield 61.6822% better accuracy. From the sample frame to 25, 50, 75,100 shown in Fig. 15, frame to 25, 100 of the test results using a standard Mean Shift occurs out of the tracking. While applying the proposed method, only the 100th frame is out of tracking while the rest of the frames are very clear. It shows that the proposed method produces better tracking performance as compared to methods utilizing the standard Mean Shift.

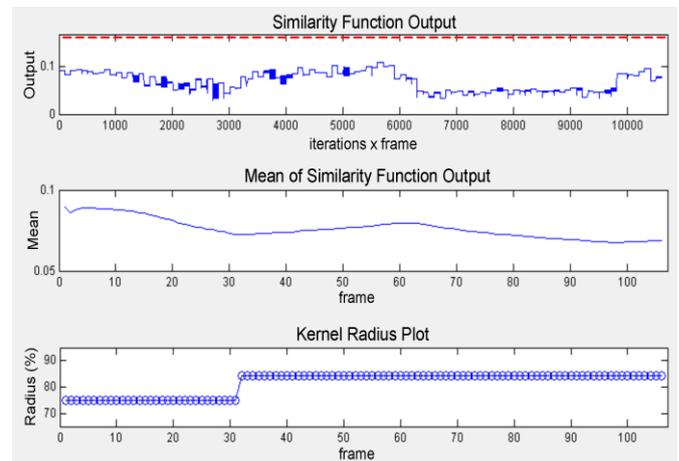


Fig. 12. Similarity Function Output, Mean of Similarity Function Output and Kernel Radius Plot respectively of Video 1 License Plate Tracking.

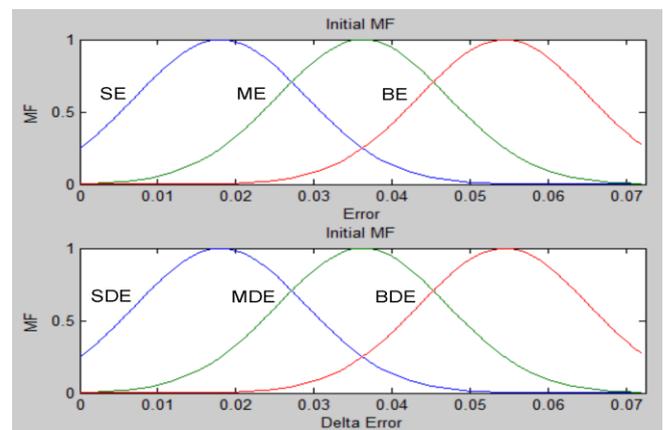


Fig. 13. Initial Membership Function of Error and Delta Error for Adaptive Fuzzy.

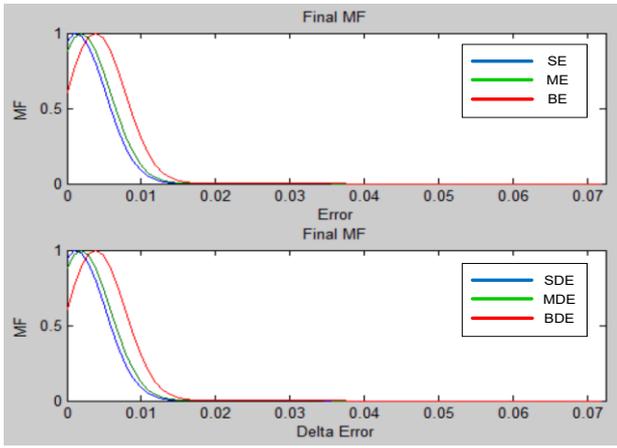


Fig. 14. Final membership Function of Error and Delta Error for Adaptive Fuzzy.



Fig. 15. The result of the process of tracking Video 1 left using the Static Gaussian kernel and the right of using Adaptive Fuzzy Gaussian kernel. Tracking results (from the top) of frames 25, 50, 75,100 are displayed.

For Video 2, the following results are obtained. The simulation results are shown in Fig. 16 through Fig. 19. The standard Mean Shift generates percentage accuracy of object tracking of 76.3780%, while the Mean Shift with adaptive fuzzy Gaussian kernel yields 80.3150% (better accuracy). From the frame samples as shown in Fig. 19, on the 90th frame, the test results using Standard Mean Shift appear out of the track very clearly. It shows that the proposed method produces better tracking performance as compared to methods utilizing the standard Mean Shift.

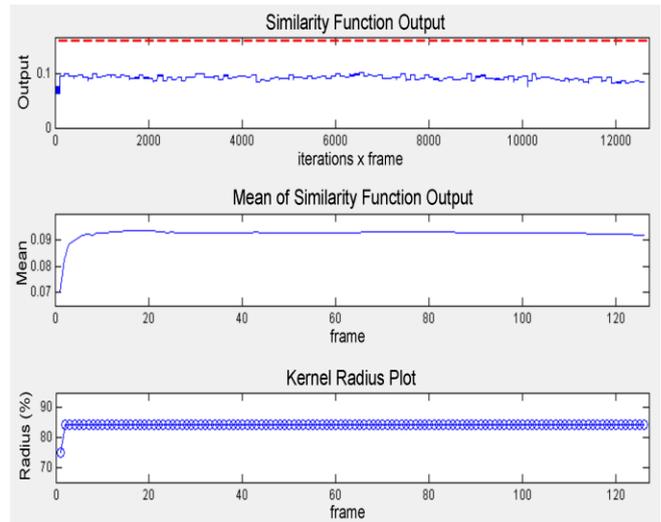


Fig. 16. Similarity Function Output, Mean of Similarity Function Output and Kernel Radius Plot respectively of Video 2 License Plate Tracking.

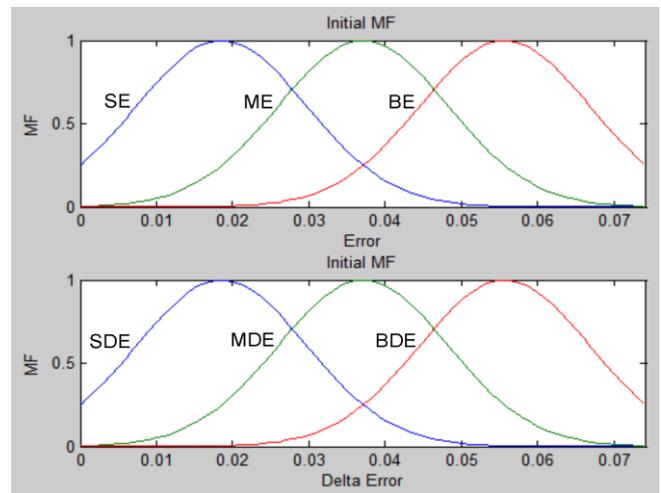


Fig. 17. Initial Membership Function of Error and Delta Error for Adaptive Fuzzy.

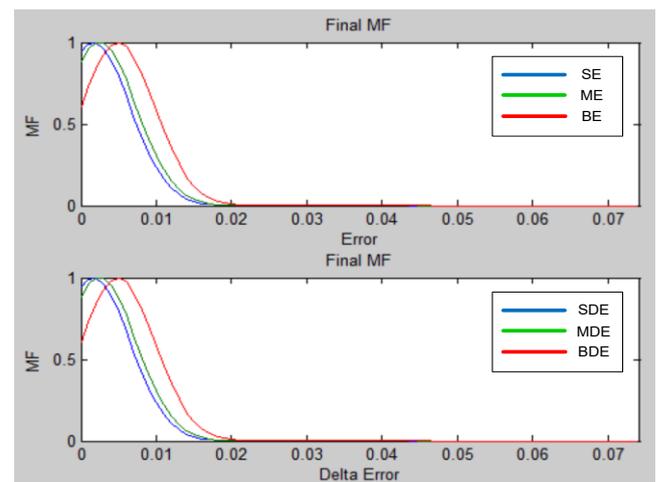


Fig. 18. Final membership Function of Error and Delta Error for Adaptive Fuzzy.



Fig. 19. The result of the process of tracking Video 2 left using the Static Gaussian kernel and the right of using Adaptive Fuzzy Gaussian kernel. Tracking results (from the top) of frames 30, 60, 90,120 are displayed.

For Video 3, the following results are obtained, as shown in Fig. 20 through Fig. 23. The standard Mean Shift generating percentage accuracy of object tracking is 62.9630%, while the Mean Shift with adaptive fuzzy Gaussian kernel yields 64.8148% (better accuracy). From these results, the difference is not significant. From the sample frame to 27, 54, 81, 108 as shown in Fig. 23, the tracking results are almost the same. Only the frame to 108, of the test results using a standard Mean Shift, occurs out of the tracking. It shows that the proposed method produces better tracking performance as compared to methods utilizing the standard Mean Shift.

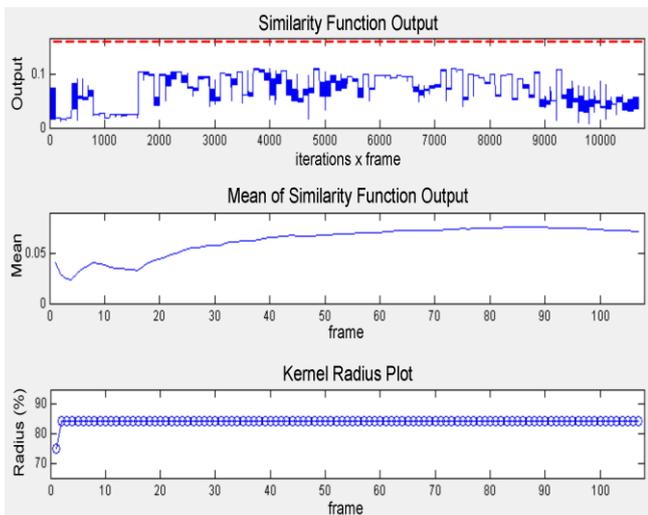


Fig. 20. Similarity Function Output, Mean of Similarity Function Output and Kernel Radius Plot respectively of Video 3 License Plate Tracking.

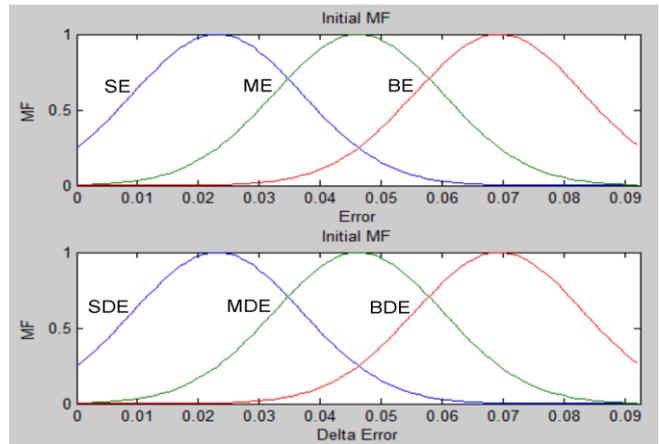


Fig. 21. Initial Membership Function of Error and Delta Error for Adaptive Fuzzy.

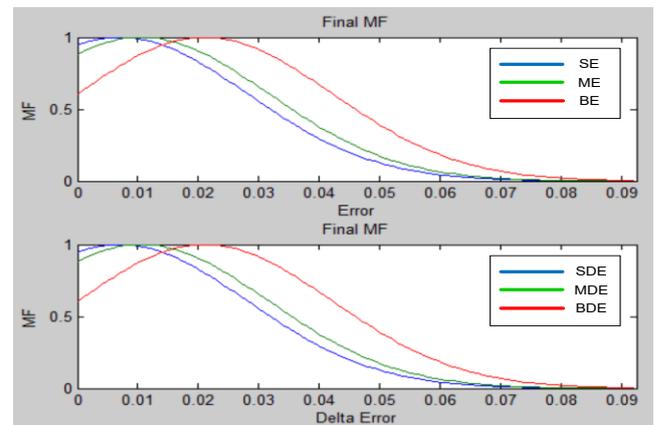


Fig. 22. Final membership Function of Error and Delta Error for Adaptive Fuzzy.



Fig. 23. The result of the process of tracking Video 3 left using the Static Gaussian kernel and the right of using Adaptive Fuzzy Gaussian kernel. Tracking results (from the top) of frames 27, 54, 81,108 are displayed.

For Video 4, the following results are obtained, as shown in Fig. 24 through Fig. 27.

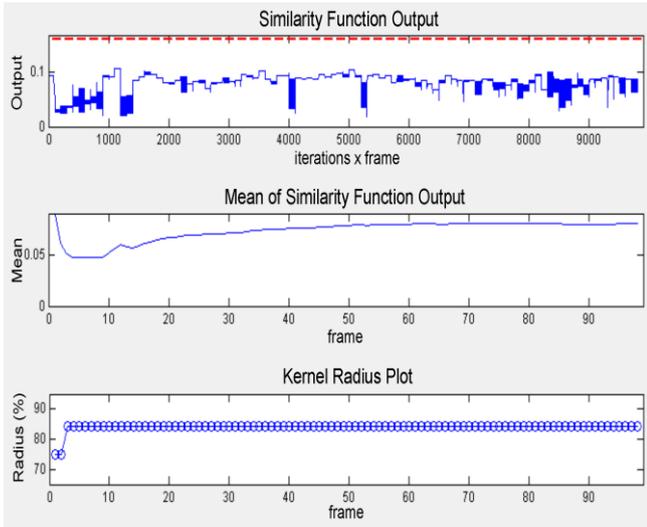


Fig. 24. Similarity Function Output, Mean of Similarity Function Output and Kernel Radius Plot respectively of Video 4 License Plate Tracking.



Fig. 27. The result of the process of tracking Video 4 left using the Static Gaussian kernel and the right of using Adaptive Fuzzy Gaussian kernel. Tracking results (from the top) of frames 24, 48, 72, 96 are displayed.

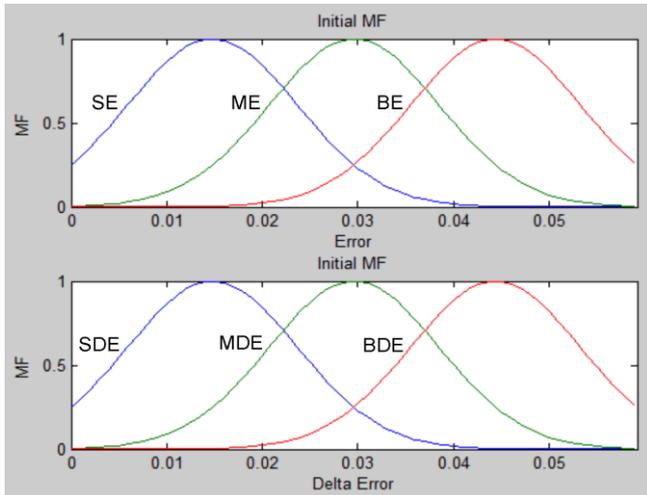


Fig. 25. Initial Membership Function of Error and Delta Error for Adaptive Fuzzy.

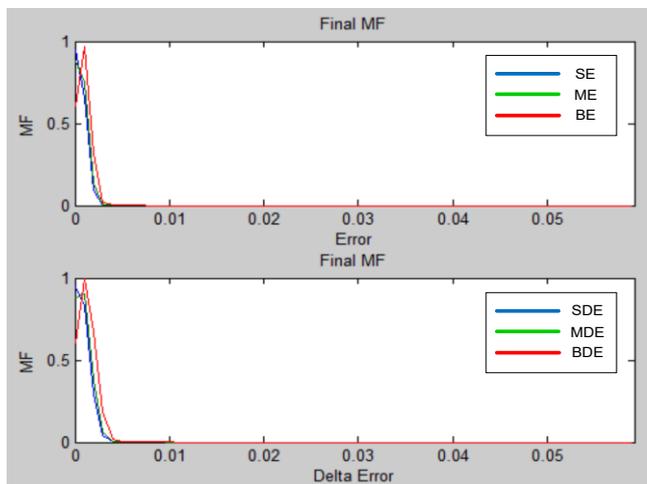


Fig. 26. Final membership Function of Error and Delta Error for Adaptive Fuzzy.

From the test results Video 4, in sample frames to 24, 48, 72, 96, it can be seen the tracking results using the Mean Shift standards (left), 3 of the four samples of the object license plate occur lost track. While the tracking results using the improved Mean Shift with adaptive fuzzy Gaussian kernel (right), all license plate objects can be tracked properly. It shows that the proposed method produces better tracking performance as compared to methods utilizing the standard Mean Shift.

Determination of tracking accuracy, it is considered Percentage Accuracy of Object Tracking (PAOT) defined in Eq. (37).

$$PAOT = \frac{\sum_1^N \text{Object on track in Frame}}{\sum_1^N \text{Frame}} \times 100\% \quad (37)$$

The test results are summarized in Table III.

Video	MS using Static Gaussian Kernel	MS using Adaptive Fuzzy Gaussian Kernel
1	50.4673	61.6822
2	76.3780	80.3150
3	62.9630	64.8148
4	28.2828	57.5758
Average	54.5228	66.0970

From the test results, the improved Mean Shift with adaptive fuzzy Gaussian kernel gave the average percentage of 66.10% tracking accuracy. With the overall trial of four videos (100%) in a "head to head," it has superior performance compared to the standard Mean Shift.

V. CONCLUSION

The paper proposed an improved Mean Shift using adaptive fuzzy Gaussian kernel for video-based Indonesian vehicle license plate tracking. The accuracy of tracking was determined by using the Percentage Accuracy of Object Tracking (PAOT). The experimental results showed that the Improved Mean Shift using Adaptive Fuzzy Gaussian Kernel provided a better average of Percentage Accuracy of Object Tracking based on the trial of four videos in comparison with the Standard Mean Shift.

ACKNOWLEDGMENT

The first author is grateful to the doctoral fellowship program funded by the Directorate General of Higher Education, Ministry of Research and Education, Indonesia.

REFERENCES

- [1] M. A. A. Aziz, J. Niu, X. Zhao, and X. Li, "Efficient and Robust Learning for Sustainable and Reacquisition-Enabled Hand Tracking," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 945–958, Apr. 2016.
- [2] V. Bastani, L. Marcenaro, and C. S. Regazzoni, "Online Nonparametric Bayesian Activity Mining and Analysis From Surveillance Video," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2089–2102, May 2016.
- [3] J. Ding, Y. Huang, W. Liu, and K. Huang, "Severely Blurred Object Tracking by Learning Deep Image Representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 319–331, Feb. 2016.
- [4] J. Meng, J. Yuan, J. Yang, G. Wang, and Y. P. Tan, "Object Instance Search in Videos via Spatio-Temporal Trajectory Discovery," *IEEE Trans. Multimed.*, vol. 18, no. 1, pp. 116–127, Jan. 2016.
- [5] S. F. Razavi, H. Sajedi, and M. E. Shiri, "Integration of colour and uniform interlaced derivative patterns for object tracking," *IET Image Process.*, vol. 10, no. 5, pp. 381–390, 2016.
- [6] A. S. Silva, F. M. Q. Severgnini, M. L. Oliveira, V. M. S. Mendes, and Z. M. A. Peixoto, "Object Tracking by Color and Active Contour Models Segmentation," *IEEE Lat. Am. Trans.*, vol. 14, no. 3, pp. 1488–1493, Mar. 2016.
- [7] J. Sun, S. Zhang, and L. Zhang, "Object Tracking With Spatial Context Model," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 727–731, May 2016.
- [8] K. R. Reddy, K. H. Priya, and N. Neelima, "Object Detection and Tracking -- A Survey," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 418–421.
- [9] S. P. Sahoo and S. Ari, "Automated human tracking using advanced mean shift algorithm," in *Communications and Signal Processing (ICCSP), 2015 International Conference on*, 2015, pp. 789–793.
- [10] M. Owczarek, P. Baranski, and P. Strumillo, "Pedestrian tracking in video sequences: A particle filtering approach," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, 2015, pp. 875–881.
- [11] S. Chan, X. Zhou, and S. Chen, "Online learning for classification and object tracking with superpixel," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 1758–1763.
- [12] Y. Yang, X. Wang, J. Wu, H. Chen, and Z. Han, "An improved mean shift object tracking algorithm based on ORB feature matching," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, 2015, pp. 4996–4999.
- [13] E. Maggio and A. Cavallaro, *Video tracking: theory and practice*. John Wiley & Sons, 2011.
- [14] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *Circuits Syst. Video Technol. IEEE Trans.*, vol. 23, no. 2, pp. 311–325, Feb. 2013.
- [15] Y. Song, S. Li, C. Zhu, S. Jiang, and H. Chang, "Invariant foreground occupation ratio for scale adaptive mean shift tracking," *IET Comput. Vis.*, vol. 9, no. 4, pp. 489–499, 2015.
- [16] W. Yu, X. Tian, Z. Hou, Y. Zha, and Y. Yang, "Multi-scale mean shift tracking," *IET Comput. Vis.*, vol. 9, no. 1, pp. 110–123, 2015.
- [17] S. Zhang, C. Wang, S. C. Chan, X. Wei, and C. H. Ho, "New Object Detection, Tracking, and Recognition Approaches for Video Surveillance Over Camera Network," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2679–2691, May 2015.
- [18] N. Li, D. Zhang, X. Gu, L. Huang, W. Liu, and T. Xu, "An improved mean shift algorithm for moving object tracking," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, 2015, pp. 1425–1429.
- [19] S. B. Setyawan, D. Purwanto, and R. Mardiyanto, "Visual object tracking using improved Mean Shift algorithm," in *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2015, pp. 1–7.
- [20] M. N. Mokti and R. A. Salam, "Hybrid of Mean-shift and median-cut algorithm for fish segmentation," in *2008 International Conference on Electronic Design*, 2008, pp. 1–5.
- [21] G. q. Li and Z. z. Chen, "A hybrid algorithm for detecting contour of moving object based on merging Mean Shift and GVF Snake model," in *2011 4th International Congress on Image and Signal Processing*, 2011, vol. 3, pp. 1287–1291.
- [22] R. He and Y. Zhu, "A Hybrid Image Segmentation Approach Based on Mean Shift and Fuzzy C-Means," in *2009 Asia-Pacific Conference on Information Processing*, 2009, vol. 1, pp. 105–108.
- [23] E. Maggio and A. Cavallaro, "Hybrid Particle Filter and Mean Shift tracker with adaptive transition model," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, vol. 2, pp. 221–224.
- [24] A. S. Khattak, G. Raja, N. Anjum, and M. Qasim, "Integration of Mean-Shift and Particle Filter: A Survey," in *2014 12th International Conference on Frontiers of Information Technology*, 2014, pp. 286–291.
- [25] Z. Linlin, F. Baojie, L. Benjin, and T. Yandong, "A Hybrid Tracking Method Based on Active Contour and Mean Shift Algorithm," in *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, 2009, pp. 70–73.
- [26] M. Azghani, A. Aghagolzadeh, S. Ghaemi, and M. Kouzehgar, "Intelligent modified mean shift tracking using genetic algorithm," in *2010 5th International Symposium on Telecommunications*, 2010, pp. 806–811.
- [27] Y. W. B. G. Zhongda Tian Shujiang Li, "Priority Scheduling of Networked Control System Based on Fuzzy Controller with Self-tuning Scale Factor," *IAENG International Journal of Computer Science*, vol. 44, no. 3, pp. 308–315, 2017.
- [28] A. M. N. A. A. A. Henry N. Afiq A. Dahlan and Sumeru, "Indoor Temperature Control and Energy Saving Potential of Split Unit Air Conditioning System using Fuzzy Logic Controller," *IAENG International Journal of Computer Science*, vol. 43, no. 4, pp. 402–405, 2016.
- [29] V. Jayalakshmi and T. A. Razak, "Trust Based Power Aware Secure Source Routing Protocol using Fuzzy Logic for Mobile Adhoc Networks," *IAENG International Journal of Computer Science*, vol. 43, no. 1, pp. 98–107, 2016.
- [30] C.-T. L. S. P. J. S. Mukesh Prasad Dong-Lin Li and S. Joshi, "Designing Mamdani-Type Fuzzy Reasoning for Visualizing Prediction Problems Based on Collaborative Fuzzy Clustering," *IAENG International Journal of Computer Science*, vol. 42, no. 4, pp. 404–411, 2015.
- [31] Khalil Khiabani and S. R. Aghabozorgi, "Adaptive Time-Variant Model Optimization for Fuzzy-Time-Series Forecasting," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 107–116, 2015.
- [32] X. Hong, "Pattern Recognition," *Mach. Intell.*, 2012.
- [33] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *Ann. Math. Stat.*, vol. 27, no. 3, pp. 832–837, 1956.
- [34] E. Parzen, "On Estimation of a Probability Density Function and Mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [35] R. Gutierrez-Osuna, "Pattern Analysis," *CSCE 666*, 2013.
- [36] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.
- [37] K. Chen, S. Fu, K. Song, and C. G. Jhun, "A Meanshift-based imbedded computer vision system design for real-time target

- tracking,” in *Computer Science Education (ICCSE), 2012 7th International Conference on*, 2012, pp. 1298–1303.
- [38] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975.
- [39] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [40] Y. Ukrainitz and B. Sarel, “Mean Shift Theory and Applications,” *Weizmann Inst. Sci.* http://www.wisdom.weizmann.ac.il/~vision/courses/2004_2/files/mean_shift/mean_shift.ppt, 2004.
- [41] K. Chen, S. Fu, K. Song, and C. G. Jhun, “A Meanshift-based imbedded computer vision system design for real-time target tracking,” 2012, pp. 1298–1303.
- [42] S. Thirumuruganathan, “Introduction To Mean Shift Algorithm,” 2010.
- [43] S. Bernhardt, “Mean-shift video tracking.” 2016.
- [44] L.-X. Wang, “A Course in Fuzzy Systems and Control,” *London Prentice-Hall Int. Inc.*, 1997.



Basuki Rahmat, He is a lecturer at bachelor degree in Universitas Pembangunan Nasional Veteran Jawa Timur. He received the bachelor degree in Instrumentation Physics from Institut Teknologi Sepuluh Nopember Surabaya in 1995. He received a master degree in Instrumentation and Control from Institut Teknologi Bandung, in 2000. Currently, he is a Ph.D. candidate in Electrical Engineering at Institut Teknologi Sepuluh Nopember, Surabaya. His research interests are an intelligent system, soft computing, image and video processing, intelligent control, MATLAB, PHP, Python and Delphi Programming.



Endra Joelianto, received the bachelor degree in Engineering Physics from Institut Teknologi Bandung (ITB), Indonesia in 1990. He received Ph.D. in Engineering, from The Australian National University (ANU), Australia in 2002. Currently, he is the staff of Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung (ITB), Indonesia and Research Professor at Centre for UnManned System Studies (CentrUMS), ITB, Indonesia. His research interests are Hybrid/Discrete Event Control Systems, Advanced Control, Embedded Control Systems and Intelligent Systems. He is a member of IEEE.



I Ketut Eddy Purnama, received the bachelor degree in Electrical Engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia in 1994. He received his Master of Technology from Institut Teknologi Bandung, Bandung, Indonesia in 1999. He received a Ph.D. degree from the University of Groningen, Netherlands in 2007. Currently, he is the staff of Electrical Engineering Department of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests are Data Mining, Medical Image Processing, and Intelligent System.



Mauridhi Hery Purnomo, received the bachelor degree from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia in 1985. He received his M.Eng., and Ph.D. degrees from Osaka City University, Osaka, Japan in 1995, and 1997, respectively. He has joined ITS in 1985 and has been a Professor since 2003. His current interests include intelligent system applications, image processing, medical imaging, control, and management. He is a Member of IEEE and INNS.