International Journal of Communication & Information Technology (CommIT) http://msi.binus.ac.id/commit/
Vol. 7 No. 2 Oktober 2013, pp. 53-64

CAMSHIFT IMPROVEMENT WITH MEAN-SHIFT SEGMENTATION, REGION GROWING, AND SURF METHOD

Ferdinan[1]; Yaya Suryana[2]
[1]PT Consulting Services Indonesia
Jl. Jendral Sudirman Kavling 28, Jakarta 10210, Indonesia, i.am.kyofu@gmail.com
[2]Centre for the Telecomunication and Information Technology,
Agency for the Assessment and Application of Technology,
Jakarta, Indonesia, yaya.suryana@gmail.com

Abstract: CAMSHIFT algorithm has been widely used in object tracking. CAMSHIFT utilizes color features as the model object. Thus, original CAMSHIFT may fail when the object color is similar with the background color. In this study, we propose CAMSHIFT tracker combined with mean-shift segmentation, region growing, and SURF in order to improve the tracking accuracy. The mean-shift segmentation and region growing are applied in object localization phase to extract the important parts of the object. Hue-distance, saturation, and value are used to calculate the Bhattacharyya distance to judge whether the tracked object is lost. Once the object is judged lost, SURF is used to find the lost object, and CAMSHIFT can retrack the object. The Object tracking system is built with OpenCV. Some measurements of accuracy have done using frame-based metrics. We use datasets BoBoT (Bonn Benchmark on Tracking) to measure accuracy of the system. The results demonstrate that CAMSHIFT combined with mean-shift segmentation, region growing, and SURF method has higher accuracy than the previous methods.

Keywords: Object Tracking, Mean-Shift Segmentation, Region Growing, CAMSHIFT, SURF

INTRODUCTION

Object tracking is an interesting research in the field of computer vision. Object tracking is the first step of a wide range of existing applications in daily life such as robotics, security, surveillance, medical imaging, sports, etc.

Object tracking normally detects the presence of objects captured by the camera, locks the object as a target, marks it and follows the moving object. There are various methods of tracking objects, where each of the methods has advantages and disadvantages. One of the popular methods is CAMSHIFT, stands for (Continuously Adaptive Mean-Shift). Various studies of CAMSHIFT have been done to improve the performance of the object tracking.

Reference [1] proposed an improved CAMSHIFT algorithm based on the kalman filtering to solve the poor tracking ability problem in occlusions. The kalman filtering used in object tracking is an optimal estimation method with the criterion of minimal error covariance. It has the advantages such as low calculation scale and high real-time performance. However, its assumption of the linear Gauss state space model may not be consistent with the motion situation of the object in the real world.

Reference [2] utilized CAMSHIFT algorithm to track a moving vehicle in traffic by combining CAMSHIFT tracker with double difference method in static background model. The double difference method was applied to separate the background and the foreground. The result of this study showed that the method could perform a calibration automatically on moving vehicle in the traffic and was able to achieve multi object tracking using multi-object tracking CAMSHIFT.

Reference [3] proposed to use of a hue distance component. The hue distance component represented a hue value as a distance from a reference hue value, which was generated dynamically from the object histogram. In addition, this study proposed a combination of hue-distance and saturation components to generate a better back projection image. The combination of the hue-distance and saturation components could eliminate the pixels in a frame that did not match the object. Thereby, tracking in similar background to the object could be more effective.

Reference [4] proposed a better object localization using mean-shift segmentation and region growing. The mean-shift was applied to segment each part of the object and made the object reasonably homogeneous to be selected easily. After the mean-shift segmentation process, then a user had to select the important part of the object by a computer mouse. The selected position became a seed point, and a region would be extended from the selected position. By using this method, one could avoid taking the background information of the object, which was not required by the object model. In addition, they used hue distance, saturation, value component to better discriminate the object model.

The use of point features has also been widely used to improve CAMSHIFT tracker that uses only color features. SIFT (Scale Invariant Feature Transform) is an approach for detecting and extracting local feature descriptors that are invariant to scale, rotation, illumination and view point changes and was proposed by Ref. [5]. SIFT is often used for object recognition. Along with the development of research, SIFT has also been widely applied in tracking objects [6, 7]. However, the implementation of SIFT in object tracking system requires a significant computation time due to its complexity.

Instead of using SIFT, Ref. [8] proposed called SURF (Speeded Up Robust Feature) method, which was also an approach to detect and extract the features of an image, and is faster than SIFT method. Reference [9] utilized CAMSHIFT in a combination with SURF method for object tracking. They also proposed a method to judge whether the tracked object is lost. Once the target is judged lost, SURF is utilized to find it again. Thus, CAMSHIFT can retrack the lost object.

By considering the previous studies, this study intends to apply the methods proposed by Ref. [4] which are a combination of mean-shift segmentation and region growing in object localization phase, and apply the method proposed by Ref. [9] to determine whether the object is lost. Once the target is judged lost, SURF is utilized to find the lost object, so that CAMSHIFT can retrack the object.

CAMSHIFT

CAMSHIFT algorithm is a modification of the mean-shift algorithm proposed by [10]. The mean-shift algorithm is a non-parametric technique for climbing the probability density distribution to find the mode (peak) of a probability distribution [3]. In 1975, the mean-shift algorithm was first proposed by [11] and applied to pattern recognition. In 1995, Ref. [12] introduced it to the field of computer vision. The mean-shift algorithm operates on a probability distribution. The probability distribution obtained from video image sequences is always changing. Therefore, the mean shift algorithm is modified to deal with dynamically changing color probability distributions derived from video frame sequences.

To meet these needs, Ref. [10] proposed a tracking method called CAMSHIFT. The main idea was to perform a mean-shift operation on all frames in the video sequence, use the center of mass and the size of the search window obtained from the previous frame as the initial values for the next frame, and achieve object tracking by iteration process.

CAMSHIFT uses color histogram as the object model. Since the histogram of the image of the object is the probability of the color object, this algorithm may not be easily influenced by changes in the shape of the object, and CAMSHIFT is able to perform tracking partially or fully occluded object. But it depends on several conditions. In addition, CAMSHIFT has fast computational time. Thus, it can perform real-time tracking.

The procedure CAMSHIFT tracker is: (1) choose the initial size and location of the search window which contains the object we want to track. Then, calculate the color histogram of that area as the object model. (2) Make the probability distribution of the current frame using the color histogram of the object model using histogram back projection method to generate back projection image (image probability distribution). (3) Based on the probability distribution of the image, calculate the location of the center of mass (centroid) within the search window, center the search window on the center of mass, and calculate the area. (4) If search window converges (the location of the new window is moved to a distance below a pre-set threshold), then return to step 2, otherwise return to step 3 until convergence.

To calculate the center of mass using the following equation, where I (x, y) is the value of each pixel in the image of the probability distribution:

0-order moment:

$$M_{00} = \sum_x \sum_y I(x,y) \tag{1}$$

1-order moment:

$$M_{01} = \sum_x \sum_y yI(x,y) \tag{2}$$

$$M_{10} = \sum_x \sum_y xI(x,y) \tag{3}$$

Then the coordinates of the center of the search window is:

$$x_c - \frac{M_{10}}{M_{00}}, y_c - \frac{M_{01}}{M_{00}} \tag{4}$$

To determine the orientation of the probability distribution use:

2-order moment:

$$M_{20} = \sum_x \sum_y x^2 I(x,y) \tag{5}$$

$$M_{11} = \sum_x \sum_y xyI(x,y) \tag{6}$$

$$M_{02} = \sum_x \sum_y y^2 I(x,y) \tag{7}$$

$$\theta = \frac{\tan^{-1}\left(\frac{2\left(\frac{M_{11}}{M_{00}}-x_c y_c\right)}{\left(\frac{M_{20}}{M_{00}}-x_c^2\right)-\left(\frac{M_{02}}{M_{00}}-y_c^2\right)}\right)}{2} \qquad (8)$$

Since the search window is ellipse-sized, the length *l* and the width *w* is:

$$l = \sqrt{\frac{(a+c)+\sqrt{b^2+(a-c)^2}}{2}} \qquad (9)$$

$$w = \sqrt{\frac{(a+c)-\sqrt{b^2+(a-c)^2}}{2}} \qquad (10)$$

where:

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \ , b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right) \ , c = \frac{M_{02}}{M_{00}} - y_c^2$$

SURF

SIFT (Scale Invariant Feature Transform) is an image feature proposed by Lowe. It is invariant to rotation and scale. SIFT is often used in image matching, but the complexity and the data used are very large and thus require a longer computation time. SURF offers a better performance than the SIFT [8].

Just like SIFT, SURF detectors are first employed to find the interest points in an image, and then the descriptors are used to extract the feature vectors at each interest point [13]. However, instead of difference of Gaussians (DoG) filter used in SIFT, SURF uses Hessian-matrix approximation operating on the integral image to locate the interest points, which reduces the computation time drastically.

Interest Point Detection

To make a feature invariant to the scale changes, the first step is construct a scale spaces [8]. Scale space is also called Gaussian pyramid. It is used to find the key point at different scales. The scale space is divided into several levels of scale called octave. Each octave representing the filter response is obtained by performing the convolution of filter box to the input image with the larger filter size according to the size of the image to construct the pyramid scale space, like the SIFT.

In SIFT, an image is convoluted repeatedly using the Gaussian kernel and then sub-sampled in order to achieve a higher level of the pyramid [9]. This method results in that each layer relies on the previous, and thus, the complexity is very large. To reduce the computation time, SURF uses box filters as an approximation of the second partial derivative of a Gaussian.

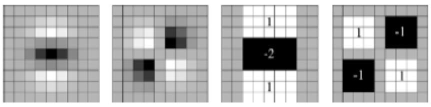

Fig. 1: From left to right: the second partial derivative of Gaussian in the direction of the y-axis and the xy axis; approach used with the filter box.
Gray area is equal to 0.

For scale invariant, SURF using Hessian matrix determinant of the localization key point is used to determine whether a point is an extreme value. Given a point $x = (x, y)$ in an image I, the Hessian metrics H (x, σ) in x at scale σ, is defined as follows:

$$H(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xx}(\mathbf{x},\sigma) \\ L_{xy}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{bmatrix}$$

Where $L_{xx}(\mathbf{x},\sigma), L_{xy}(\mathbf{x},\sigma)$, and $L_{yy}(\mathbf{x},\sigma)$ are the convolutions of the Gaussian second order partial derivatives with the image ***I*** in point ***x*** respectively.

To reduce computation time, a 9 x 9 box filter size is used as the interpretation of a Gaussian with σ = 1.2 and represents the lowest scale to compute the blob response maps, which is denoted by D_{xx}, D_{yy}, and D_{xy}. The equation is as follows:

$$det\left(H_{approx}\right) = D_{xx}D_{yy} - \left(\omega D_{xy}\right)^2 \qquad (11)$$

The weights applied to the rectangular regions are kept simple for computational efficiency. In the above equation, *ω* is a weight to balance the Hessian's determinant needed to balance the relative weights.

To localize the key point in the image and over scales, non-maximum suppression in a 3x3x3 neighborhood is applied. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Ref. [14].

Interest Point Description

Description of key point is given so that the key point descriptor vector is invariant to rotation, illumination, and view point changes [8]. To create a key point that is invariant to the rotation changes, every feature detected will be given an orientation.

SURF key point descriptor relies on the dominant orientation of the entire key point, and then components of the descriptor built. The dominant orientation calculation is based on the Haarwavelet response, which calculates Haar response on the X and Y coordinates in a region where its center is a circle with a radius 6s key point. It is by sampling at each scales, as well as the calculation of Haarwavelet response in accordance with the scale. Soon a large scale, the size of the wavelets will also be large. At this stage the use of the integral image is used to perform a quick filter. Thus, it only needs six operations to calculate the response on the x axis and y axis on each scale. The size of the Haarwavelet is the 4sand the number of vectors calculated every 60 degrees in a circle. Finally, the orientation with the largest sum of vectors is the dominant orientation. The process of determining the orientation is shown in Fig. 2.

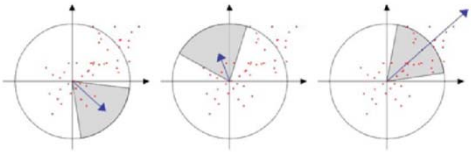

Fig. 2: Orientation Assignment

Once the dominant orientation is determined, to extract descriptors, a square window is constructed at each key point and centered around the key point. The size of the square window is 20σ. The region is split up regularly into smaller 4x4 sub-regions. This region still preserves important spatial information as the original region. For each sub-region, then compute Haar wavelet response in 5x5 space sample points. For simplicity, Haar wavelet response in horizontal direction is denoted as d_x and response in the vertical direction is denoted as d_y. To increase the robustness towards geometric deformations and localization errors, the responses d_x and d_y are first weighted with a Gaussian (σ = 3.3 s) centered at the key point.

Then the wavelet response d_x and d_y summed up over each sub-region and form a first set of entries in feature vector. This would describe the information about the polarity of the intensity changes, and also extract the sum of the absolute values of the responses$|d_x|$and$|d_y|$. Hence, each sub-region has a 4D descriptor vector **v** for its underlying intensity structure **v**, d_x, d_y, $|d_x|$, and$|d_y|$. Concatenating this for all 4x4 sub-regions results in a descriptor vector of length 64. Normalizing it, then we get the descriptor components which are invariant to scale, rotation, and translation changes of images.

Mean-Shift Segmentation

In ref [15] mean-shift segmentation is a segmentation algorithm/clustering. The aim of this algorithm is to find the local maxima of the probability density given by the observations [16]. Mean-shift segmentation smoothen the image and remove noise while preserving discontinuity [17].

Region Growing

Region growing is one of the segmentation techniques [18]. Region growing is a procedure that groups pixels or sub-regions into larger regions based on predefined criteria for growth. The basic approach is to start with a set of “seed” points and from these grow regions by appending to each seed those neighboring pixels that have predefined properties similar to the seed (such as specific ranges of intensity or color).

One of the region growing methods often used is flood fill. Based on [15] flood fill is a very useful function which is often used to mark or isolate an area in an image for further analysis or processing. In OpenCV, flood fill can also be used to generate a mask that can be used in the next process to accelerate or restrict a process of image processing where only pixels that are processed in accordance with the area indicated by the mask.

METHOD

Object Localization

Determination of the location and size of these arch windows by using a rectangle is still the same with original CAMSHIFT. It is because this step is needed to define the region of interest (ROI) of an image. After getting ROI, then do mean-shift segmentation to smoothen the image and make theme homogeneous enough to be chosen easily.

To perform the mean-shift segmentation, we use mean-shift segmentation implementation in OpenCV library. Based on [15] Initial values for the mean-shift segmentation: the spatial range (hs) = 20 and the color range (hr) = 40, and the maximum level = 2. In our experiments, these values cannot be applied. They are good for an image dimension 640x480 while the mean-shift segmentation in this study is only used in the selection of objects where ROI image size may vary. Hence, the value needs to be redefined according to the complexity of the object.

The next step is object selection by clicking on each part of the object. The click position becomes a seed point that has specific properties. From these seed points, the region grows by appending their neighbors which have similar properties to the seed. In addition, there are tolerance values of colors so that a seed can grow until reaching the edges of object parts. Region growing implementation in OpenCV generates a mask. This mask indicates the parts of the object. Each object selected by clicking on it, the mask will be combined using the OR operator (see Fig. 3).

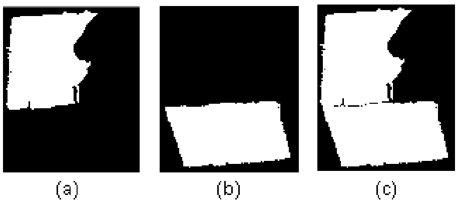

Fig. 3: Merging mask with OR operator

This method avoids the background information that is not needed for object model, while the original CAMSHIFT does not pay attention to the background information on the object, so that the background information will be calculated into the object model. In addition, mask is created for SURF. This mask will be used to restrict an area when SURF detects the feature of an image. The use of this mask is also to avoid SURF to detect background of the object. Hence, SURF only detects an area that possibility of the existence of the object in a frame.

Object Modeling

The next step is object modeling, which uses color histogram as the object model. We use HSV color model, same as the original CAMSHIFT. Original CAMSHIFT only uses the hue component, but this can be a problem in the condition where the object has similar hue to the background, so that the tracking can be fail.

Reference [19] used hue-distance and saturation components. The hue distance is a function which represents each hue value H as a distance from a reference hue value H_{ref}. The following distance function is used instead of the hue component:

$$d(H, H_{ref}) = \begin{cases} |H - H_{ref}| & if |H - H_{ref}| \leq 180^\circ \\ 360^\circ - |H - H_{ref}| & if |H - H_{ref}| > 180^\circ \end{cases} \quad (12)$$

Hue reference H_{ref} is the value which has the highest frequency obtained from the normal hue histogram. First, the hue histogram of the standard HSV color model is calculated, and then the hue histogram obtained is recalculated with the above equation.

The use of hue distance and saturation components is also used by [4]. According to [4] the use of hue distance and saturation components might sufficient for some cases, but for some other cases, the use of hue distance and saturation is not good enough to distinguish the object from its background. Value color component often gives good discrimination between objects with its background. Therefore, we use three components as [4].

We also do quantization for each histogram with a smaller bin. Based on our experiments and compared with the previous study, the use of 30 bins for hue-distance component, 9 bins for saturation, and 16 bins for value gives good result.

Color Masking

Color masking is done on every frame. Each pixel in every frame will be evaluated based on the minimum and maximum values of hue, saturation, and value components. If the pixel values are in the range of minimum and maximum values, it will be given 1, otherwise it will be given 0. The image mask is made at this stage will be combined with a back projection image. In addition, the use of mask is also used to restrict area when SURF is used to find the lost object. However, in this study, the mask is made and combined with back projection image that is not used for SURF. It is because the mask used to combine with back projection focusing on color of the object is being tracked. Hence, there are missing part of the objects in the mask which may be an important part of the object while mask is used to restrict an area when SURF detects key point should represent full part of the objects.

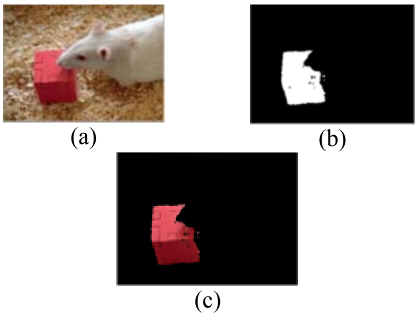

Fig. 4: (a) original image, (b) mask that used to restrict area when SURF detects feature, (c) an area which will be used by SURF for key point detection

Histogram Back projection

Histogram back projection evaluates every pixel in the frame sequence based on histogram model we have made in the object modeling phase. We do histogram back projection to hue-distance histogram, saturation histogram, and value histogram. The output of histogram back projection is back projection image which contains the probability of each pixel in the frame according to the histograms. Then the three back projection images are combined into a single back projection image using AND operator. Starting from the back projection image hue-distance, back projection image saturation, back projection image value, and the last is mask that was created in the previous phase.

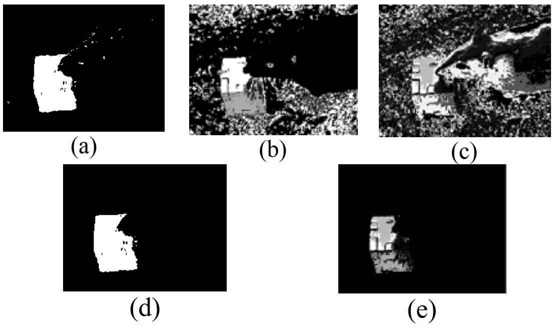

Fig. 5: (a) is back projection hue, (b) is back projection saturation, (c) is back projection value, (d) is mask, (e) the result of merging back projection image and mask with AND operator

Tracking

Once the probability distribution of the image is created, the next step is to do the tracking by operating the CAMSHIFT algorithm to update size and location of search window which will be used for the next frame.

According to [9] the object lost can usually be divided into 2 kinds of situation. First, object might be lost when the object's background has similar color to the object. Second, when the object moves too fast, the object region of 2 continuous frames do not overlap so that the search window may not converge to catch up the movement. In our experiments with original CAMSHIFT for this situation, it may cause the search window can been larged or search window moved into the background. Objects can also be declared lost if the search window area equal to 0.

To overcome this problem, [9] using Bhattacharyya distance to calculate the distance between two histograms. The formula of Bhattacharyya distance is:

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_i H_1(i) \cdot \sum_i H_2(i)}}} \quad (13)$$

An image region in a search window is calculated for hue-distance, saturation, and value histogram. Furthermore, the histogram is compared with the histograms obtained in the object modeling phase. Based on our experiments, the object lost if Bhattacharyya distance hue> 0,6 or Bhattacharyya distance saturation> 0,5 or Bhattacharyya distance value> 0,4.

Once the object is judged lost, first extract features from the image ROI and the current frame by using SURF. In addition, using a mask made in the previous phase to restrict area that represents the presence of the object when extracting SURF features. After key points have obtained, the next step is to do key point matching using implementation of FLANN (Fast Approximate Nearest Neighbor Library) and Homography in OpenCV library. This step will be obtained a new size and location of the search window. In this case, the new size and location of the search window is also a tracking phase. Then, the new size and location of these arch windows becomes an initial value for the next frame.

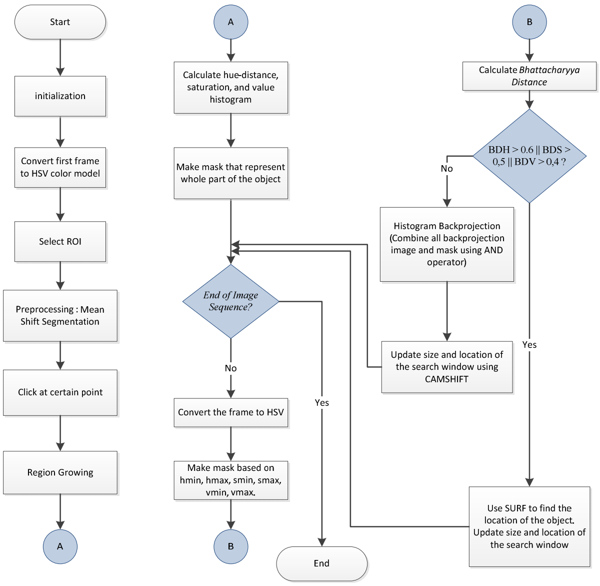
Start
initialization
Convert first frame to HSV color model
Select ROI
Preprocessing : Mean Shift Segmentation
Click at certain point
Region Growing
A
A
Calculate hue-distance, saturation, and value histogram
Make mask that represent whole part of the object
End of Image Sequence?
No
Yes
Convert the frame to HSV
Make mask based on hmin, hmax, smin, smax, vmin, vmax.
B
End
B
Calculate Bhattacharyya Distance
BDH > 0.6 || BDS > 0,5 || BDV > 0,4 ?
No
Histogram Backprojection (Combine all backprojection image and mask using AND operator)
Update size and location of the search window using CAMSHIFT
Yes
Use SURF to find the location of the object. Update size and location of the search window

Fig. 6: The proposed system diagram

Object tracking system in this study is shown in Fig. 6. The procedure of the system is: (1) Initialization of the extreme values: hue min, hue max, saturation min, saturation max, value min, and value max; (2) Choose initial size and location of the search window, which contains the object we want to track; (3) convert ROI image to HSV color model; (4) Do mean-shift segmentation continued with select the object parts by clicking on each object part; clicking the object part will perform region growing; (5) Calculate hue distance, saturation, and value histogram of the selected object parts; (6) Create mask for current frame, each pixel will be evaluated based on the extreme values that have defined in step 1; (7) convert current frame to the HSV color model; (8) do histogram back projection to make a probability distribution of the frame using the color histogram obtained in step 5; merge them with mask image obtained in step 6; (9) Calculate Bhattacharyya distance of tracked object with histograms obtained in step 5; If score Bhattacharyya distance hue > 0.5 or score Bhattacharyya distance saturation > 0.6 or score Bhattacharyya distance value > 0.4 , then the object is lost; (10) If the object is lost, find the object using SURF; update size and location of the search window if the object is not lost; update the size and location of the search window using CAMSHIFT algorithm; (11) Repeat step 6 until 10 until the last frame.

Performance Evaluation Metrics

To evaluate the performance of the object tracking, we use frame-based metrics proposed by [20]. Starting with the first frame of the test sequence, frame-based metrics are computed for every frame in the sequence. From each frame in the video sequence, first a few true and false detection and tracking quantities are computed.

True Negative (TN) is number of frames where both ground truth and system results agrees on the absence of any object. True Positive (TP) is number of frames where both ground truth and system results agrees on the presence of one or more objects, and the bounding box of at least one or more objects coincides among ground truth and tracker results. False Negative (FN) is number of frames where ground truth contains at least one object, while system either does not contain any object or none of the system's objects falls within the bounding box of any ground truth object. False Positive (FP) is number of frames where system results contains at least one object, while ground truth either does not contain any object or none of the ground truth's objects falls within the bounding box of any system object.

The calculation of TP, TN, FP, and FN can be shown as follows:

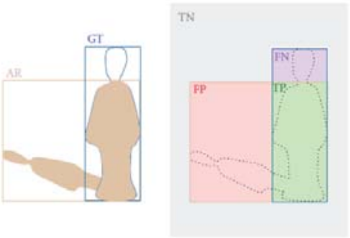

Fig. 7: Illustration of Calculation of TP, TN, FP, FN [21]

Once the above defined quantities are calculated for all the frames, the following metrics are computed:

$$\text{Tracker Detection Rate (TRDR)} = \frac{\text{TP}}{\text{TG}} \tag{14}$$

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{TP+FP}} \tag{15}$$

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP+FN}} \tag{16}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP+TN}} \tag{17}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TF}} \tag{18}$$

$$\text{Positive Prediction} = \frac{\text{TP}}{\text{TP+FP}} \tag{19}$$

$$\text{Negative Prediction} = \frac{\text{TN}}{\text{FN+TN}} \tag{20}$$

$$\text{False Negative Rate} = \frac{\text{FN}}{\text{FN+TP}} \tag{21}$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP+TN}} \tag{22}$$

There was a limitation when measuring TP, TN, FP, and FN. The measurements are performed manually by observing the bounding box of GT and the bounding box of the system result by considering the condition according to Fig. 7 for each frame. When the system reads the first frame of the video, the system will automatically pause for each frame. To go to the next frame, press any button on the keyboard.

RESULTS AND DISCUSSION

In this study, we use ground truth data to evaluate performance of the tracking system. Ground truth data are intentionally made to compare the results between system results and ground truth data. In order to perform evaluation of the proposed systems, we use data sets obtained from BoBoT (Bonn Benchmark on Tracking). There are 12 videos with dimension 320×240, and 25 fps. However, the data used in this study are only 10 with different cases (moving cam, illumination changes, scale changes, rotation changes, and similar object's background).

The tests are carried out on four systems; they are: CAMSHIFT combined with mean-shift segmentation, region growing, and SURF (CAMSHIFT + SM + RG + SURF), Original CAMSHIFT, CAMSHIFT combined with mean-shift segmentation and region growing (CAMSHIFT + SM + RG), and CAMSHIFT combined with SURF (CAMSHIFT + SURF).

Some screenshots of the tracking process are shown in Figs. 8–13.

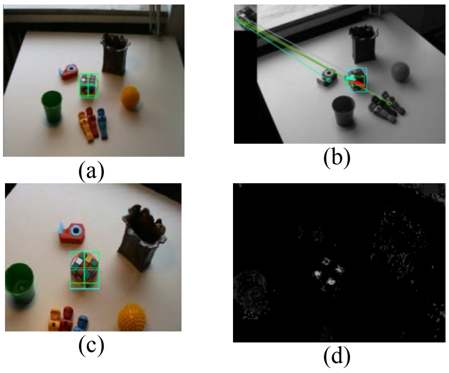
(a)
(b)
(c)
(d)

Fig 8: Tracking with the proposed system in Vid_G_rubikscube.avi video. (a) Tracking at frame 318, (b) SURF feature matching at frame 121, (c) Tracking at frame 642, (d) Back projection image at frame 642

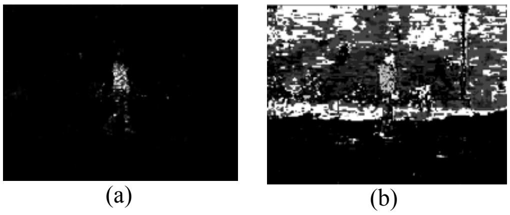

Fig 9: Tracking in Vid_D_person.avi.
(a)Back projection image produced by the proposed system at frame 627, (b) Back projection image produced by original CAMSHIFTat frame 627

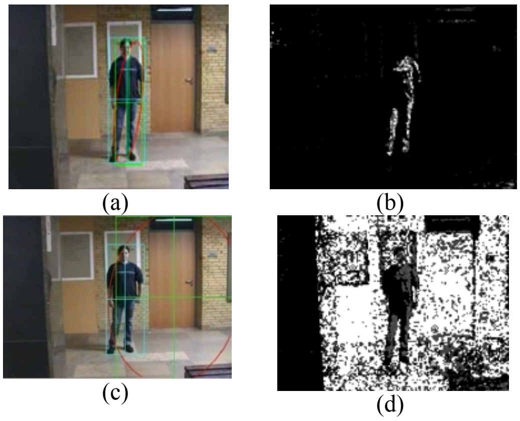

Fig 10: Tracking in Vid_E_person_partially_occluded.aviat frame 17. (a) Tracking with the proposed system;
(b) Back projection image produced by the proposed system; Tracking with original CAMSHIFT; (c) Tracking only approximate the object part; (d) Back projection image produced by original CAMSHIFT

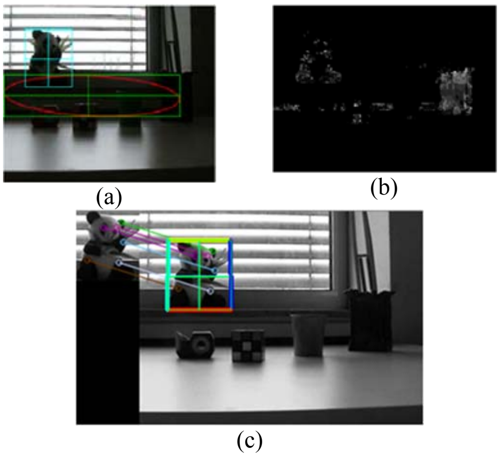

Fig 11: Tracking with the proposed system in Vid_H_panda.avi video. (a) Track wrong object at frame 114; (b) Back projection at frame 114; (c) SURF feature matching at frame 115.

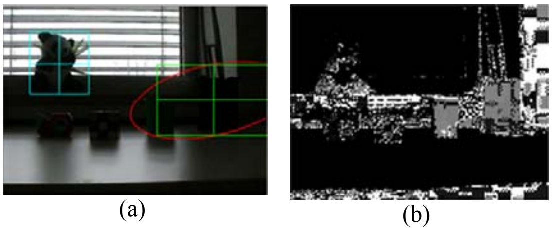

Fig 12: Tracking with original CAMSHIFT in Vid_H_panda.avi video. (a) Track wrong object at frame 114; (b) Back projection image at frame 102

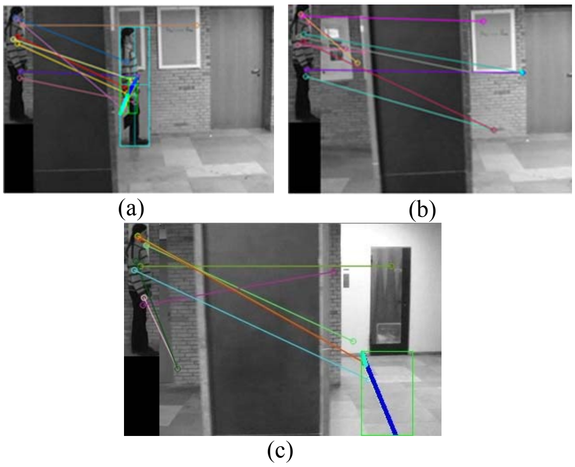

Fig 13: Tracking with the proposed system in Vid_F_person_fully_occluded.avi video. (a) SURF feature matching successfully find the lost object, (b) SURF feature matching does not find any object, (c) SURF Feature matching fails to find the object.

Figures14–22 show a comparison of each metric for each video.

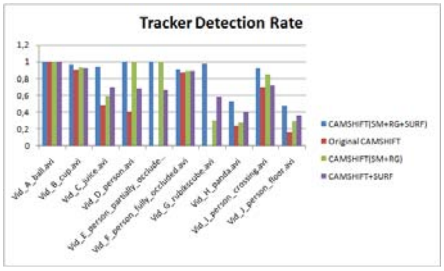

Fig. 14: Comparison of Tracker Detection Rate

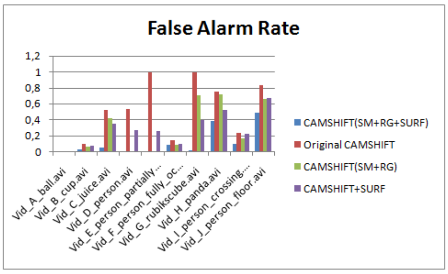

Fig. 15: Comparison of False Alarm Rate

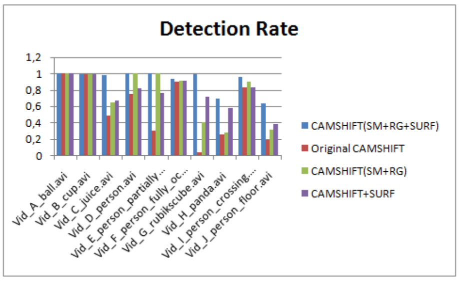

Fig. 16: Comparison of Detection Rate

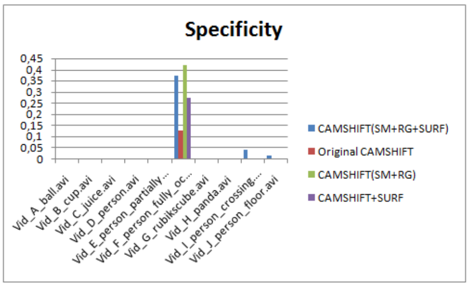

Fig. 17: Comparison of Specificity

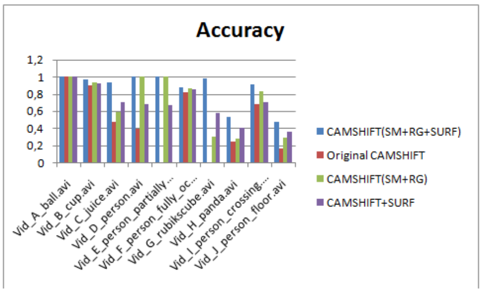

Fig. 18: Comparison of Accuracy

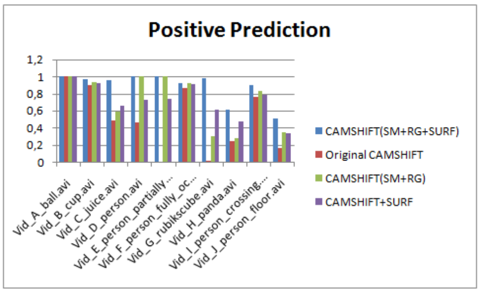

Fig. 19: Comparison of Positive Prediction

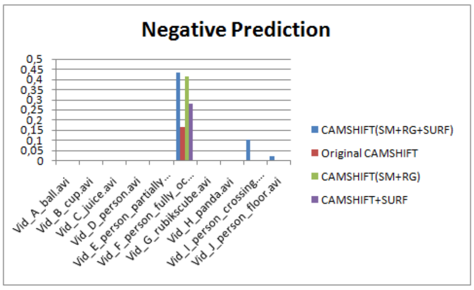

Fig. 20: Comparison of Negative Prediction

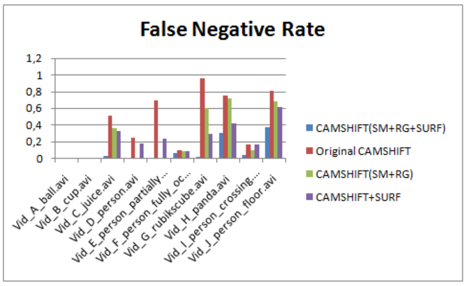

Fig. 21: Comparison of False Negative Rate

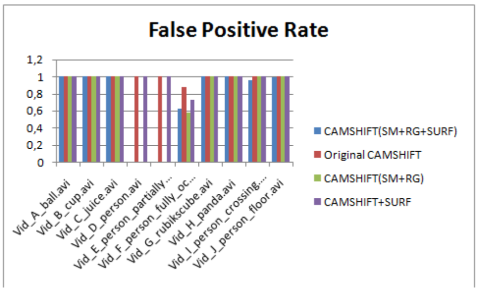

Fig. 22: Comparison of False Positive Rate

Table 1 below shows the average result of performance evaluation using frame-based metrics.

Table 1: Average value of frame-based metrics

Metrics	CAMSHIFT + SM + RG + SURF	Original CAMSHIFT	CAMSHIFT + SM + RG	CAMSHIFT + SURF
Tracker Detection Rate	87,14%	47,95%	71,30%	69,18%
False Alarm Rate	11,32%	50,73%	28,02%	28,27%
Detection Rate	91,94%	57,38%	74,61%	76,76%
Specificity	4,27%	1,25%	4,21%	2,75%
Accuracy	86,66%	47,24%	70,89%	68,56%
Positive Prediction	88,68%	49,34%	71,99%	71,73%
Negative Prediction	5,53%	1,63%	4,14%	2,80%
False Negative Rate	8,07%	42,62%	25,39%	23,24%
False Positive Rate	75,76%	98,75%	75,79%	97,26%

This study focuses on the accuracy of the object tracking system. As we can see in Table1, CAMSHIFT combined with mean-shift segmentation, region growing, and SURF has the highest value. Thus, the difference of accuracy between systems is an improvement obtained.

CAMSHIFT use color histogram (color feature) as the object model. To achieve good result of object tracking using CAMSHIFT, it requires good back projection image (probability distribution image). The use of mean-shift segmentation and region growing for object localization proposed by [4] is a great way to obtain the color information of the object that we want to be tracked because the calculation to generate back projection image based on color histogram. However, the use of mean-shift segmentation and region growing requires more time to select the object parts since the user should specify which area of an object that to be obtained.

Determination of object lost proposed by [9] using Bhattacharyya distance is a simple and effective enough to judge whether the tracked object is lost. In previous study [9], the calculation of Bhattacharyya distance only uses hues histogram. In this study, we use three histograms to judge whether the object is lost. The use of three components is done since similar background object may have different saturation or value. And hence, with this three components could better assess whether the tracked object is lost. However, the use of three components make it more sensitive to illumination changes, so that SURF will of then be used in cases of frequent illumination changes.

Use of SURF method to find the lost object is very helpful CAMSHIFT tracker to retrack the object. However, tracking error of then occurs in SURF feature matching. Color masking that used at object selection to restrict area when SURF detects feature is needed to prevents SURF detect features of object's background. It is very important thing, because it is one key success of SURF feature matching. In our experiment, the use of this mask can speed up the computation time of SURF, because only certain areas are used. Thus, SURF feature matching can be more effective. However, making mask by utilizing extreme value has not been good enough to eliminate the pixels that do not represent the object.

From the test that has done, we have known the capabilities of our object tracking system. Ref. [22] does a qualitative comparison between object tracking methods. Qualitative comparison of kernel trackers is obtained based on: tracking single or multiple objects, ability to handle occlusion, requirement of training, type of motion model, and requirement.

Table 2: Qualitative Comparison of Geometric Model-Based Trackers (Init. denotes initialization. #: number of objects, M: multiple objects, S: single object respectively, A: affine or homography, T: translational motion, S: scaling, R: rotation, P: partial occlusion, F: full occlusion. Symbols √ and × respectively denote if the tracker requires or does not require training or initialization.) [22]

	#	Motion	Training	Occ.	Init
Simple template matching	*S*	*T*	×	*P*	√
Mean-shift [Comaniciu et al. 2003]	*S*	*T+S*	×	*P*	√
KLT [Shi and Tomasi 1994]	*S*	*A*	×	*P*	√
Appearance Tracking [Jepson et al. 2003]	*S*	*T+S+R*	×	*P*	√
Layering [Tao et al. 2002]	*M*	*T+S+R*	×	*F*	×
Bramble [Isard and MacCormick 2001]	*M*	*T+S+R*	√	*F*	×
EigenTracker [Black and Jepson 1998]	*S*	*A*	√	*P*	√
SVM [Avidan 2001]	*S*	*T*	√	*P*	√

Based on qualitative comparison, our object tracking system in this study using CAMSHIFT tracker, and combined with mean-shift segmentation, region growing, and SURF. The proposed system only tracks single object (S), the motion model of the proposed system can deal with translation motion (T), scale changes (S), rotation changes (R), and does not need training (×), can track object although only partially (P), and even object that is fully occlusion but depending on the condition of the object's color. In "Vid_I_person_crossing" video, the object several times is covered by the objects that block or interfere with the object, but if the object whose cover has a color that is very similar to the color of the object, the system can track the object even though the object is fully closed. The proposed system requires initialization (√) which requires the color histogram as object models and determination of the extreme values is very important to make a mask.

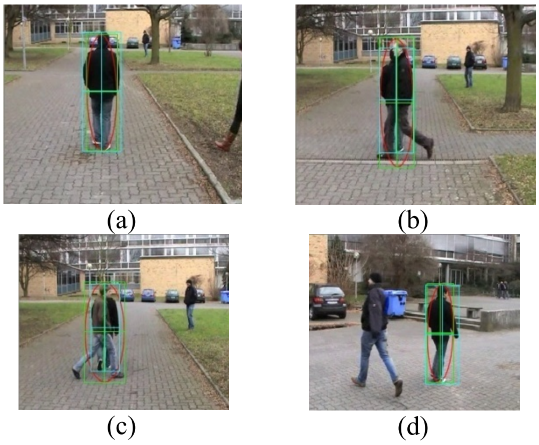

Fig 23: Tracking with proposed system in Vid_I_person_crossing.avi video.
(a) Tracking at frame 93,
(b) Tracking at 328 (fully occluded),
(c) Tracking at frame 432 (fully occluded),
(d) Tracking at frame 950.

However, one important thing in this case is the proposed system in this research can judge whether the tracked object is lost. And if the object is lost, the system can find the lost object, which in this study using SURF, so CAMSHIFT can retrack the object. Besides, a better method is needed to restrict an area when SURF detects features. Therefore, our future work could focus on method that use for restrict an area of an image to raise the accuracy of the SURF feature matching.

CONCLUSION

In this study, we propose an improved CAMSHIFT tracker with mean-shift segmentation, region growing, and SURF. Mean-shift segmentation and region growing method are applied in the object localization phase to avoid taking background information of the object. SURF is used to extract feature and SURF feature matching is done to find location of the object when the object is lost.

The results demonstrates that the use of mean-shift segmentation, region growing, and SURF can improve accuracy of object tracking system. Improvement value acquired based on the difference of accuracy between the proposed system and original CAMSHIFT is 39.42%. Improvement value acquired based on the difference of accuracy between the proposed system and CAMSHIFT combined with mean-shift segmentation and region growing is 15.77%. Improvement value acquired based on the difference of accuracy between the proposed system and CAMSHIFT combined with SURF is 18.1%.

REFERENCES

[1] W. Jiang-tao and Y. Jing-yu, "Object tracking based on kalman-mean shift in occlusions," *Journal of System Simulation*, pp. 4216–4220, 2007.

[2] J. Xia, J. Wu, H. Zhai, and Z. Cui, "Moving vehicle tracking based on double difference and CAMShift," *Proceedings of the 2009 International Symposium on Information Processing*, pp. 29–32, 2009.

[3] M. K. Chouhan, R. Mishra, and D. D. Nitnawwre, "Movable object tracking by using mean shift method with adjusted background histogram," *International Journal of Advanced Research in Computer Science and Software Engineering*, pp. 16–19, 2012.

[4] P. Hidayatullah and H. Konik, "CAMSHIFT improvement on multi-hue and multi-object tracking," *International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 143–148, 2011.

[5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, 60(2), pp. 91–110, 2004.

[6] Q. Xuena, L. Shirong, and L. Fei, "Kernel-based target tracking with multiple features fusion," *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference (CDC/CCC)*, pp. 3112–3117, 2009.

[7] K. Du, Y. Ju, Y. Jin, G. Li, Y. Li, and S. Qian, "Object tracking based on improved MeanShift and SIFT," *2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 2716–2719, 2012.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding 110(2008)*, pp. 346–359, 2007.

[9] J. Li, J. Zhang, Z. Zhou, W. Guo, B. Wang, and Q. Zhao, "Object tracking using improved Camshift with SURF method," *International Workshop on Open-Source Software for Scientific Computation (OSSC)*, pp. 136–141, 2011.

[10] G. R. Bradski and S. Clara, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technology Journal, 2nd Quarter*, 1998.

[11] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transaction on Information Theory*, pp. 32–40, 1975.

[12] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transaction Pattern Analysis and Machine Intelligence*, pp. 790–799, 1995.

[13] G. Du, F. Su, and A. Cai, "Face Recognition using SURF features," *Proc. of SPIEE* , pp. 1–7, 2009.

[14] M. Brown and D. Lowe, "Invariant Features from Interest Point Groups," *BMVC*, 2002.

[15] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, California: O'Reilly Media, 2008.

[16] H. Zhou, J. Wu, and J. Zhang, *Digital Image Processing: Part II.* Ventus Publishing, 2010.

[17] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 603–619, 2002.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd Ed., New Jersey: Prentice-Hall, 2002.

[19] J. A. Corrales, P. Gil, F. A. Candelas, and F. Torres, “Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System,” *In Proceedings Book of 40th International Symposium on Robotics*, p. 107, AER-ATP, Barcelona, Spain: Asociación Española de Robótica y Automatización Tecnologías de la Producción, 2009.

[20] F. Bashir and F. Porikli, “Performance Evaluation of Object Detection and Tracking Systems,” *MITSUBISHI ELECTRIC RESEARCH LABORATORIES*, 2006.

[21] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. S. Loos, and M. Merkel, “A Review and Comparison of Measures for Automatic Video Surveillance Systems,” *EURASIP*, 2008.

[22] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: a survey,” *The ACM Computing Surveys*, 38(4), pp. 118–124, 2006.