# Resolving Occlusion in Multi-Object Tracking through Integrated Fuzzy Similarity Measure

Rahmatri Mardiko
Faculty of Computer Science University of Indonesia
Kampus UI Depok 16424 - Indonesia
mardiko@gmail.com

M. Rahmat Widyanto
Faculty of Computer Science University of Indonesia
Kampus UI Depok 16424 - Indonesia
widyanto@cs.ui.ac.id

***Abstract*--** **In multi-object tracking, occlusion is a situation where part of an object is covered by another object or any structure in the video scene. It is a very common problem in multi-object tracking for real world video scenes and is a cause for poor tracking performance. Considering its significance and inevitability, this problem has been a subject of numerous papers about multi-object tracking. In this paper, a method for occlusion handling based on fuzzy approach is proposed. Fuzzy techniques are used here as they can deal with uncertainty and imprecision which are inherent in image/video processing. The method consists of feature extraction, fuzzy feature representation, merge-split event detection, and track resolution. The main contribution of this paper is in the use of fuzzy similarity measure together with fuzzy integral for resolving object tracks after occlusion. The similarity measure is performed separately on color, texture, and shape after representing them as fuzzy features. Then, fuzzy integral combines them to calculate the overall similarity value. Experimental result shows that with moderately fast computational time, the proposed method can resolve occluded tracks accurately even in difficult situations. This result also shows the promising applicability of fuzzy approach for future automated video surveillance research.**

***Index Term*--** **Multi-object tracking, occlusion handling, fuzzy similarity measure, fuzzy integral.**

## 1. INTRODUCTION

Automated video surveillance is one of the most active research areas within computer vision community. The rapid development in this area is motivated mainly by the rising demand of its wide range of applications. To mention some of them, with automated video surveillance we can improve people security, perform congestion analysis in urban traffic, and help visually impaired people. The goal of this research area is to obtain high level description about the captured scenes in videos.

To achieve this goal, several processes are involved including environment modeling, foreground extraction, object classification, tracking, and high level analysis [1]. Environment modeling and foreground extraction together are performed to separate the static (background) and the moving parts (foreground) of the video. The extracted foregrounds are then analyzed and some are recognized as objects. Classification is performed to distinguish between different classes of objects such as pedestrians, vehicles, or animals. When an object appears in a video scene, its trajectory is recorded through object correspondence in consecutive frames. This trajectory maintenance describes the role of tracking. Finally, ones can get the description of things and situations in the video by analyzing object movements, trajectories, or recognizing its identity, behavior, and many others. Among these steps, this paper is mainly concerned with object tracking. However, since the environment modeling and foreground extraction are required in the method, they are also discussed.

### 1.1 Occlusion Problem and Analysis

In real world video scenes, objects especially the moving ones are frequently cluttered and interactions among them are inevitable. It could potentially degenerate tracking accuracy as it is difficult to locate individual object correctly in a video scene when the object is occluded. Moreover, when a group of objects splits into its individual objects or other group of objects, the tracking task becomes more complicated since it should be able to resolve the object tracks. This problem is well-known as occlusion problem.

There are many techniques for solving occlusion problem. Generally, they can be divided into two major groups [2]: Merge-Split (MS) approach and Straight-Through (ST) approach. In MS approach, when an occlusion occurs, the information with regard to the occluded objects is frozen and a new track is created for the newly formed group. When the group splits or an object in the group leaves, the track is returned as it was before occlusion by measuring its similarity with the frozen tracks related to the group.

Whilst in ST approach, during the occlusion, the region of the formed group is segmented to obtain regions of individual objects. Compared to the former, this approach can achieve more accurate tracking because even in the middle of occlusion, the objects can be located precisely. But the drawback is that the computational cost becomes substantially more expensive. Moreover, in case the objects are fully occluded (the whole object is covered) or the occlusion occurs for long duration, the track could be lost. It could happen because the ST approach regards a region

as an object by its area and its continuity over time. If a region related to an object is difficult to extract or only the small portion of the object extracted over a long period of time, the object is considered lost.

In this study, MS approach is preferred for implementation for its robustness against full and long occlusion and efficient computational cost.

The performance of MS methods depends mainly on two things: (1) merging-splitting event detection and (2) similarity measure after occlusion. One of common approaches for detecting merging and splitting is by using bounding box of the objects [3-4]. When two or more bounding boxes are overlapped, an occlusion event seems to occur. If so, the tracking task should freeze the individual object tracks and create a new track for the group. When the group splits, similarity measure is performed to match the object based on its information before occlusion. The tracking task should find the most similar frozen track to each object.

### 1.2 Previous Works

Since occlusion is a major problem in multi-object tracking, numerous works have been carried out to solve it. In this section, several previous researches are discussed to show how the problem has been treated. The discussion below is primarily concerned with merging-splitting event detection and occlusion resolution.

Javed and Shah [4] used a distance matrix to establish object-track correspondence as well as to detect merging or splitting. If a track does not correspond to any object and the bounding box of the track is likely to overlap with other object, a merging event has possibly occurred. Distance matrix and overlapping bounding box are also used by Yang [5] and Senior [6] for merging-splitting event detection. A different method was developed by Takala and Pietikäinen [7]. Instead of merely using the bounding boxes, they surrounded the boxes with circles and used the circles to detect merging-splitting event.

The overlapping bounding boxes method provides a strong detection power. However, in certain cases, it does not work well because the box does not represent the real object shape and orientation. It typically happens when the objects are skewed due to the position of the camera. Bounding ellipses are considered more powerful in general cases. But it is more difficult to determine whether two region are overlapped each other in ellipse shape than in rectangle shape. So when choosing the bounding shapes, there is a trade-off between the detection accuracy and the cost.

When a group of objects split into individual objects, the tracking task should find the most similar track to each object. In doing this, it performs similarity measure based on the features used. McKenna et al [3] proposed a method based on color histogram similarity to resolve the tracks after occlusion. Working with RGB color histogram as the object feature, the method applies histogram intersection [8] as the measure of resemblance. The RGB color histogram is also used by Yang et al [5] in their work. Instead of histogram intersection, Kullback-Leibler (KL) distance is used as similarity measure.

Despite strong discrimination power of color as feature, the robustness of the methods that use color as the only feature is questionable. In many situations, a single descriptor cannot perform effectively when the feature is very similar.

Multiple features are supposed to give richer descriptor and hence, more accurate result. Takala and Pietikäinen [7] took into account texture feature as well as motion in addition to color. The color is represented by correlogram [9] together with histogram, the texture is represented by local binary pattern [10], and the motion by speed and direction. Their experiment showed that using more features give more accurate results. Undoubtedly, the more features used, the more computational cost the method requires. The choice of optimal features depends usually on the specific requirement of the application, the available resources, and the expected performance.

### 1.3 Fuzzy Approach

In this paper, an alternative method to solve occlusions by incorporating fuzzy approach is proposed. Fuzzy techniques provide more intuitive expressions to perceive real world values by using membership functions. One of the most important features of using fuzzy approach is its fault tolerance. With this capability, they can solve problems that comes with uncertainties and imprecision in nature. In image processing field, uncertainties appear in all levels: preprocessing, representation, interpretation, etc as described in [11]. This important feature brings fuzzy logic a wide acceptance in image processing field and has been extensively applied in many applications.

In region based object tracking, which is the subject of this work, uncertainty and imprecision exist in foreground extraction stage and hence, forwarded to the next processing step: features extraction and representation. This fact is the main reason of applying fuzzy logic to solve occlusion problem besides the inherent fuzziness of image processing applications. Fuzzy approach is expected to be more robust and give more accurate result. In the proposed method, fuzzy approaches are applied in features extraction and representation, similarity measures, and similarity aggregation to obtain total similarity values.

The remaining part of this paper is structured as follows: In section 2 the tracking method with its processing stages are described. Feature extraction and representation are described in section 3. The fuzzy similarity measure that is

used in the proposed method is discussed in section 4. Fuzzy integral as the aggregation method is discussed in section 5. Experimental results are reported in section 6. In section 7, computational time is analyzed. Finally conclusions are derived in section 7.

## 2. TRACKING METHOD

In this section, the tracking method is briefly described together with the occlusion handling algorithm. The overall tracking system is depicted in Figure 1.

The first stage in the system is foreground extraction. It attempts to acquire pixels that belong to the moving objects in the video. The resulting foreground pixels are then analyzed to obtain the connected components which are called blobs. Object correspondence between the objects and the existing tracks is performed by using blob position and size. The shape features are extracted immediately from the blobs while color and texture features are extracted by using the blobs as masks. Each object has a model that contains information about the object features, size, position, and speed. Once the correspondence is established, the models are updated. Next, occlusion is handled by performing merging-splitting event detection and resolving the object tracks immediately after a group splits. Details about these processes are described in the following subsections.

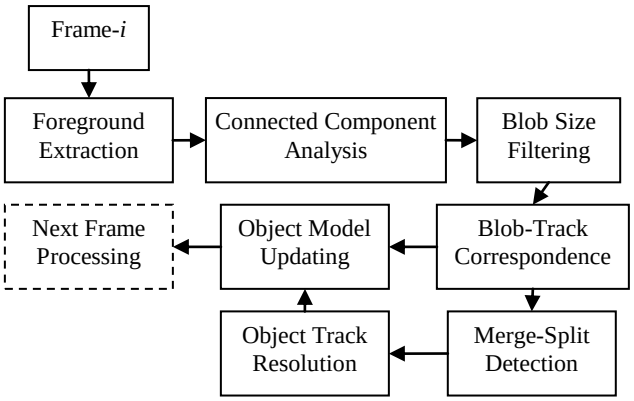


Fig. 1. Structure of The Tracking System

### 2.1 Foreground Extraction

Foreground is the area in video which is occupied by objects. In region based object tracking, the foreground should be extracted and separated from the background. This process is well known as background subtraction. Various background subtraction methods have been developed in literatures [12]. They differ in the way of modeling pixels in video frames. Among these methods, Adaptive Gaussian Mixture Models (GMM) [13] is one of which has a better accuracy among others and run moderately fast. Furthermore, shadow removal capability is added to the method so it can achieve better accuracy [14].

In adaptive GMM method, each pixel is modeled as a mixture of $k$ gaussian distributions. This model is updated in time direction to adapt with changes in video sequence. When a particular pixel is examined in frame $t$, its value is matched to the model to decide whether the pixel belongs to background. The matching process takes into account the mean and standard deviation of each gaussian distribution. The matched and non-matched pixels are considered as background and foreground respectively.

In situations with strong lumninance, shadows are often falsely classified as objects. For accurate further analysis, it is essential to obtain only the areas that belong to the objects. In [14], shadow removal is performed by separating a pixel into its contrast and luminance components. If the chromatic and luminance value of the pixel are below some predetermined threshold, it is considered as part of shadows.

Connected component analysis is applied to the resulting foreground pixels to obtain groups of pixels that are connected each other. These groups of pixels are called blobs. The blobs of size smaller than a predetermined threshold are filtered out as they are regarded as noises. Finally, object regions are represented by the remaining blobs. These regions are then used for extracting the features (shape, color, texture) of the objects.

### 2.2 Object Correspondence

Object correspondence is a main task in object tracking. It maintains identities of objects in consecutive frames. In a particular time $t$, there exist $m$ tracks which have been maintained until frame $t$-1 and $n$ blobs which have been extracted from current frame $t$. The task is to establish correspondence between the tracks and the blobs. It is performed by calculating minimum distance between each pair of track and object. Let $X$, $V$, $S$, and $\Delta S$ denote position, velocity, size, and size difference of an object respectively, the distance between track $i$ and blob $j$ is calculated as follows

$$d(i,j) = \left| X_i - \left( X_j + V_j \right) \right| + \left| S_i - \left( S_j + \Delta S_j / S_j \right) \right|. \quad \textbf{(1)}$$

As can be seen in the equation 1, the distance calculation takes into account the position as well as the size of the object. One-one correspondence is established by taking the pair $(i,j)$ that yields the minimum distance value. Based on this result, any information related to each track is updated in the following way

$$T_i = \rho T_i + (1-\rho) T_{i-1}, \quad \textbf{(2)}$$

where $T$ denotes any information (feature values) related to the tracks, $\rho$ is learning rate, indicates the extent to which the model adapt with the new values.

Once the object-track correspondence is established, one of the following conditions applies:

(1) all tracks and blobs are corresponded each other,

(2) there exists a blob which does not correspond to any of the tracks, or

(3) there exists a track which does not correspond to any of the blobs.

In case (2) there are two possibilities, either a new object entered the video or a spliting event occurred. So, a check for newly entering object and splitting event detection is called. In case (3), there are also two possibilities, either the object that is supposed to correspond to this track exited from the video or an occlusion occurred. A check for object exiting and merging event detection is called. Merging and splitting event detection are described in more details in the following subsection.

### 2.3 Merging and Splitting Detection

As stated in the introduction, the performance of MS approach depends on merging and splitting detection beside similarity measure. Thus, it is important to ensure that when objects merge or split, the tracking system can notice the events correctly. The overlapping bounding box approach [4,5,6] is used in the proposed method. The merging detection procedure is illustrated in Figure 2.

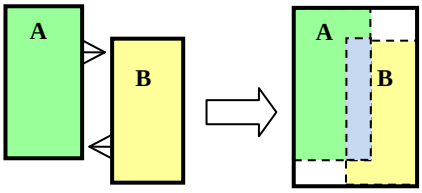


Fig. 2. Merging event detection

In case a track does not correspond to any one of the blobs, its predicted bounding box is calculated based on the previously recorded bounding box and speed. This calculation is formally defined by

$$\hat{B}_j^t = B_j^{t-1} + V_j \quad \textbf{(3)}$$

where $\hat{B}_j^t$ and $B_j^{t-1}$ denote the predicted and the previously recorded bounding box respectively, and $V_j$ denotes the speed of the object. The predicted bounding box is then checked if it is overlapped with any other objects. Note that, the bounding box is defined as a vector containing its four corner positions, so the above calculation is equivalent to vector translation.

Next, if a merging event is detected, occlusion handling is performed by firstly, freezing the tracks related to occluded objects and secondly, creating a new track for the group. As a result, the group is considered as an individual object. The frozen tracks are kept as long as the occlusion last.

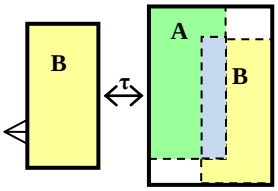


Fig. 3. Splitting event detection

Splitting event detection begins when a blob does not correspond to any existing tracks. If it is not a newly entering object, it is then checked whether there is any group around this blob within a specified distance threshold. If there is such a group, it is evident that the object has just left the group. Occlusion is resolved by measuring the similarity between the object and each frozen track related to this group. Here, fuzzy similarity measure (described in section 4 and 5) is incorporated in calculating the objects similarity by their color, texture, and shape features. The resulting similarity value is used to determine the most probable track for the object.

## 3. Fuzzy Feature Extraction and Representation

Features are used as descriptor of an object so it can be distinguished with others. Generally, the more features used the more accurate an object is represented. But too many features can cause what is called as "curse of dimensionality", a situation in which some features reduce the significance of others. In the other hand, there is a trade-off between the number of features and the computational cost. The selection of features requires analyses about the specific requirement of the application, the available computational resources, and the expected performance. This analysis is domain specific. Most automated video surveillance applications are required to be capable of providing real time analyses with respect to the captured scenes.

In multi-object tracking with occlusion handling, object features are necessary for resolving object-track correspondence by measuring their similarities once a group splits. Multiple features are used in the proposed method in order to provide more information and give more accurate results. The method takes into account color, shape, and texture features of objects. The efficiency is considered in determining the features used.

### 3.1 Feature Extraction

The color feature is extracted in HSV color space. HSV color space is chosen because it is more intuitive and similar to human perception than RGB color space. This property is helpful in applying fuzzy labels for different prominent colors based on human perception. The fuzzy labels define the membership functions that will be used to build fuzzy histogram. Only the hue and value components are used for representing color. These two components are

powerful enough to recognize color as suggested in [15]. With fewer components, time complexity and memory usage are also reduced. Instead of using joint histogram, the fuzzy histograms are built separately for hue and value component. This approach allows a more efficient representation than joint histogram and helps achieving better time performance.

The texture feature is represented by Local Binary Pattern [10] histogram. LBP is chosen for its good accuracy and speed. This feature has successfully been applied in [7]. To extract the feature, an image needs to be converted to grayscale first. The local binary pattern for each pixel in the image is calculated as follows: (1) differences between the pixel and each of 8 neighbor pixels in certain radius are calculated in counter clock-wise order; (2) the difference values are then binarized by applying a threshold value yielding a bit string of length 8; (3) the bit string is then converted to decimal number. Based on LBP value for each pixel, LBP histogram is built for the whole image.

The previously extracted foregrounds are used as masks in calculating the histograms both for color and texture. This masking provides more accurate feature extraction of the objects.

Shape is usually represented as complex features which are extracted from object contour. However, in real world video scenes, objects belonging to the same class are typically difficult to distinguish by shapes. For instance, human contour shape is very similar to each other. In this situation, shape feature is more useful for differentiating objects of different classes with obvious different shapes. Besides, extracting complex shape feature from countour is computationally expensive and not well-suited for real time application. For this reason, aspect ratio and size of the blob (in pixels) are chosen. These features are simple yet are powerful to distinguish between people, cars, and bikes. Combined with speed, these features had also been successfully used in a previous work [16].

### 3.2 Fuzzy Feature Representation

The advantage of fuzzy approach is that it can express uncertainty and vagueness of measurement explicitly. It allows partial truth values as human being does, so the problem is perceived in more intuitive way. In this study, extracted color, texture, and shape features are transformed into fuzzy representation.

The fuzzy color representation follows the method suggested in [17] with fewer color labels as used in [15]. The hue component is defined by 9 membership functions and the value component by 8 membership functions as depicted in Figure 4. As depicted in the figure, the representation allows gradual difference of membership values. It makes the feature more robust against slight difference in color values caused by noises or illumination change.

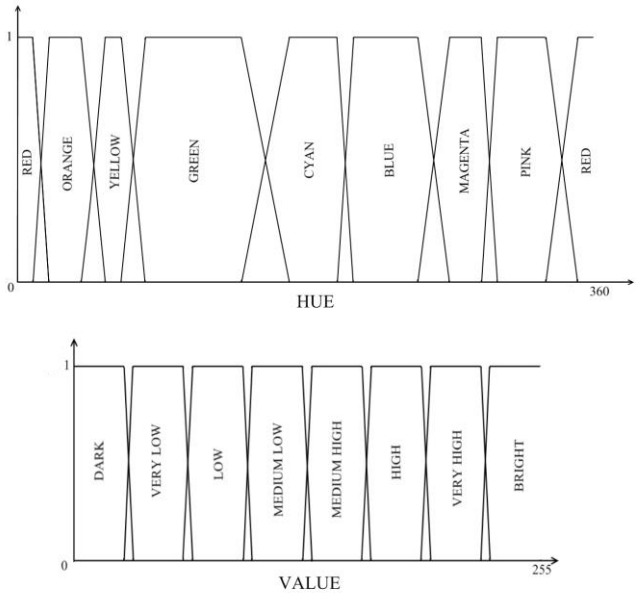


Fig. 4. Fuzzy color space for hue and value components

The similar approach is applied for fuzzy representation of local binary pattern. Since the bit string is of length 8, there are $2^8$ = 256 different values. The whole range 0-255 different values are divided into 17 partitions each of which has the same length except for the first and the last partition. There is no exact way for dividing this range and it is part of experiment. The partition is accepted as long as it express the fuzziness which is represented by gradual change in the values. Other may use smaller number of partitions with wider range of each and vice versa.

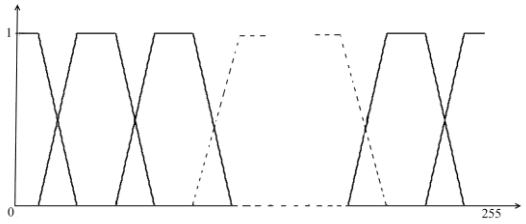


Fig. 5. Fuzzy representation of local binary pattern values

The shape feature which is combination of aspect ratio and object size are represented by fuzzy predicates. The predicates are Low, Medium, and High for aspect ratio and Small, Medium, and Large for object size. For each predicate, a membership function is defined and the feature values is determined by evaluating the raw values ageinst these functions. Hence, the shape feature consists of six values, one for each fuzzy predicates.

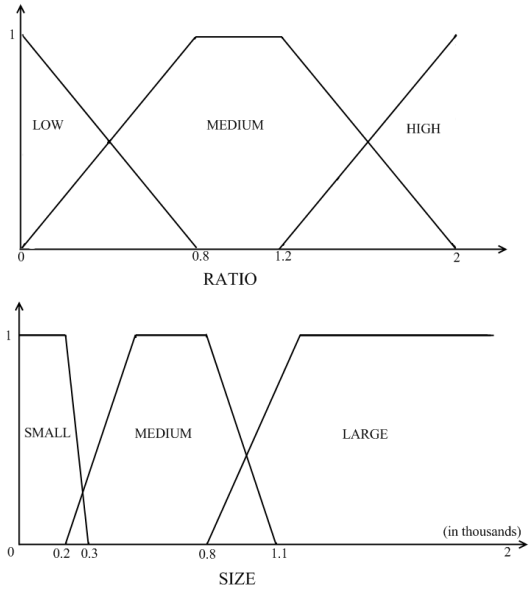


Fig. 6. Fuzzy predicates for shape feature (ratio and size)

## 4. FUZZY SIMILARITY MEASURE

Similarity measure is a common problem in image processing field, particularly for object recognition and content based image retrieval applications. Various techniques have been developed to calculate similarity between objects or features. With fuzzy approach, similarity measures could be calculated from (1) the membership values of predefined membership functions; or (2) the membership functions derived from the data [18].

In [19], a fuzzy similarity measure is developed which is called Fuzzy Feature Contrast Model (FFCM). FFCM is formulated based on Tversky's similarity concept. The main idea behind this model is that the more common features between two objects the more similar they are and the more different features between them the less similar they are. In set theory, the common and different features can be interpreted as set intersection and set difference respectively. The idea is clearly shown as the equation

$$S(A,B) = f(A \cap B) - f(A - B) - f(B - A), \quad \textbf{(4)}$$

where $S(A,B)$, $f(A \cap B)$, $f(A - B)$ denote the similarity, the intersection, and the difference between A and B respectively. In fuzzy set theoretic approach, intersection is commonly represented by min(*A*,*B*) and set difference is defined in the paper as max(*A*-*B*,0).

The original FFCM as proposed in the paper is as follows

$$S = X - (\alpha.Y + \beta.Z). \quad \textbf{(5)}$$

where *X*, *Y*, and *Z* denote A∩B, A-B, and B-A respectively. $\alpha$ and $\beta$ determine the asymmetry property of the model when $\alpha \neq \beta$. In this case, it is assumed that *Y* and *Z* are symmetric, hence $\alpha = \beta$. In [20], different yet having similar properties models are described,

$$S_{Jaccard} = \frac{X}{X + Y + Z}, \quad \textbf{(6)}$$

$$S_{Dice} = \frac{2X}{2X + Y + Z}, \quad \textbf{(7)}$$

$$S_{Jaccard} = \frac{X}{X + Y + Z}. \quad \textbf{(8)}$$

Compared to the original FFCM (eq. 5), these models can avoid values less than zero when $Y + Z$ is greater than $X$. It is an advantage since it is difficult to interpret negative similarity values.

For histogram representation, the values *X*, *Y*, *Z* are obtained by performing the operations for each pair of bin values in same indexes and then summing the results. While for the shape feature, the operation is performed on the fuzzy predicates. These operations are expressed as follows

$$X = \sum_i \min\left(\mu(A_i), \mu(B_i)\right), \quad \textbf{(9)}$$

$$Y = \sum_i \max\left(\mu(A_i) - \mu(B_i), 0\right), \quad \textbf{(10)}$$

$$Z = \sum_i \max\left(\mu(B_i) - \mu(A_i), 0\right), \quad \textbf{(11)}$$

where $\mu(X)$ denotes the membership degree of value *X* in the fuzzy histogram bin for color and texture, or in the fuzzy predicates for shape.

To obtain the best suited model for the tracking problem in this study, an experiment is conducted to compare their performance. Sample object images including pedestrians, cars, and bikes are cropped from video frames so that each object is represented by five images. A simple content based image retrieval is built where each image in database is used as query. Query accuracy is calculated as the number of images corresponding to the same object as query image in top five query result. The total accuracy is obtained by averaging all query accuracy. Note that in the experiment, the performance is not measured for shape features since in this method shape feature is used to distinguish classes of objects and not the object itself. The results showed that the three modified versions perform better than the original FFCM. Among these three models, $S_{Jaccard}$ has the advantage that it is more simple in calculation than the other two models. So, in the proposed method $S_{Jaccard}$ formula is used to perform fuzzy similarity measure for color and texture. For the color feature, similarity calculation is performed individually on hue and value fuzzy histogram and then combined by weighted

average method. For the shape feature, original FFCM model is preferred since the experiment in [19] showed that the model performs well to measure similarity between shapes that are represented by fuzzy predicates.

## 5. AGGREGATING SIMILARITIES

After obtaining individual similarity values of color, texture, and shape features, the next step is aggregating them to obtain the total similarity value. There exist several common techniques for aggregation, one of which is arithmetic sum or averaging. While it is very easy for implementation, the drawback of this technique is that it cannot distinguish the degree of importance for each component. In this case, color, texture, and shape features are supposed to have different discrimination powers. Weighted arithmetic sum seems to be promising solution, but it has some limitation in expressing joint degree of importance of the components. Fuzzy approach provides a more sophisticated tool.

Fuzzy approach provides aggregation operations that are useful in multicriteria decision making. Among these operations, fuzzy integral is used in the proposed method as the aggregation operator [21]. Not only it allows expressing individual degree of importance, fuzzy integral allows expressing the interaction among them. It is very useful in many applications for real problem solving, since the operator is more intuitive and more similar to human perception. Fuzzy integral operators are developed based on the concept of fuzzy measures which are described below. Let $X = \{x_1,...,x_n\}$ be a set of criteria and $\mathrm{P}(X)$ be the power set of $X$.

Definition 1. A fuzzy measure on the set X of criteria is a set function $\mu : \mathrm{P}(X) \to [0,1]$, satisfying the following axioms:

(i) $\mu(\phi) = 0$, $\mu(X) = 1$,

(ii) if $A \subset B \subset X$ then $\mu(A) \le \mu(B)$.

There are two variants of fuzzy integral. The first is Sugeno integral and the second is Choquet Integral. Based on preliminary experiments, Choquet integral is chosen because it seemed to give better performance in the proposed method.

Definition 2. Let $\mu$ be a fuzzy measure on $X$. The Choquet integral of a function $f : X \to [0,1]$ with respect to $\mu$ is defined by:

$$C_\mu(f(x_1),...,f(x_n)) = \sum_{i=1}^{n} (f(x_{(i)}) - f(x_{(i-1)}))\mu(A_i) \qquad \textbf{(12)}$$

where $\cdot_{(i)}$ indicates that the indices have been permuted so that $0 \le f(x_{(i)}) \le \ldots \le f(x_{(n)})$, and $A_{(i)} = \{x_{(i)}, \ldots, x_{(n)}\}$.

In this case, $X$ is a set of similarity values for all features. The fuzzy measure $\mu$ is defined so that the expected properties that describe the degree of importance and interaction among similarities are fulfilled. In the following, $S_{COLOR}$, $S_{TEXTURE}$, $S_{SHAPE}$ denote the similarity value of color, texture, and shape respectively.

The individual degree of importance for color ($\mu(\{S_{COLOR}\})$) and texture ($\mu(\{S_{TEXTURE}\})$) are determined based on experiments while for shape ($\mu(\{S_{SHAPE}\})$) is determined to be small. When the color is similar, it is more probable that the texture would be similar so the joint degree of importance is supposed to be less than the sum of individual values, hence $\mu(\{S_{COLOR}, S_{TEXTURE}\}) < \mu(\{S_{COLOR}\}) + \mu(\{S_{TEXTURE}\})$. The joint degree of color and shape is determined based on the observation that when both color and shape of two objects are similar, it gives a stronger evidence that they are the same object, hence $\mu(\{S_{COLOR}, S_{SHAPE}\}) > \mu(\{S_{COLOR}\}) + \mu(\{S_{SHAPE}\})$. The same idea is applied to determine the joint degree of texture and shape, so $\mu(\{S_{TEXTURE}, S_{SHAPE}\}) > \mu(\{S_{TEXTURE}\}) + \mu(\{S_{SHAPE}\})$. The joint degree for all features equals 1 by definition, hence $\mu(\{S_{COLOR}, S_{TEXTURE}, S_{SHAPE}\}) = 1$. Experiments and tuning are required to determine the exact values of these fuzzy measures.

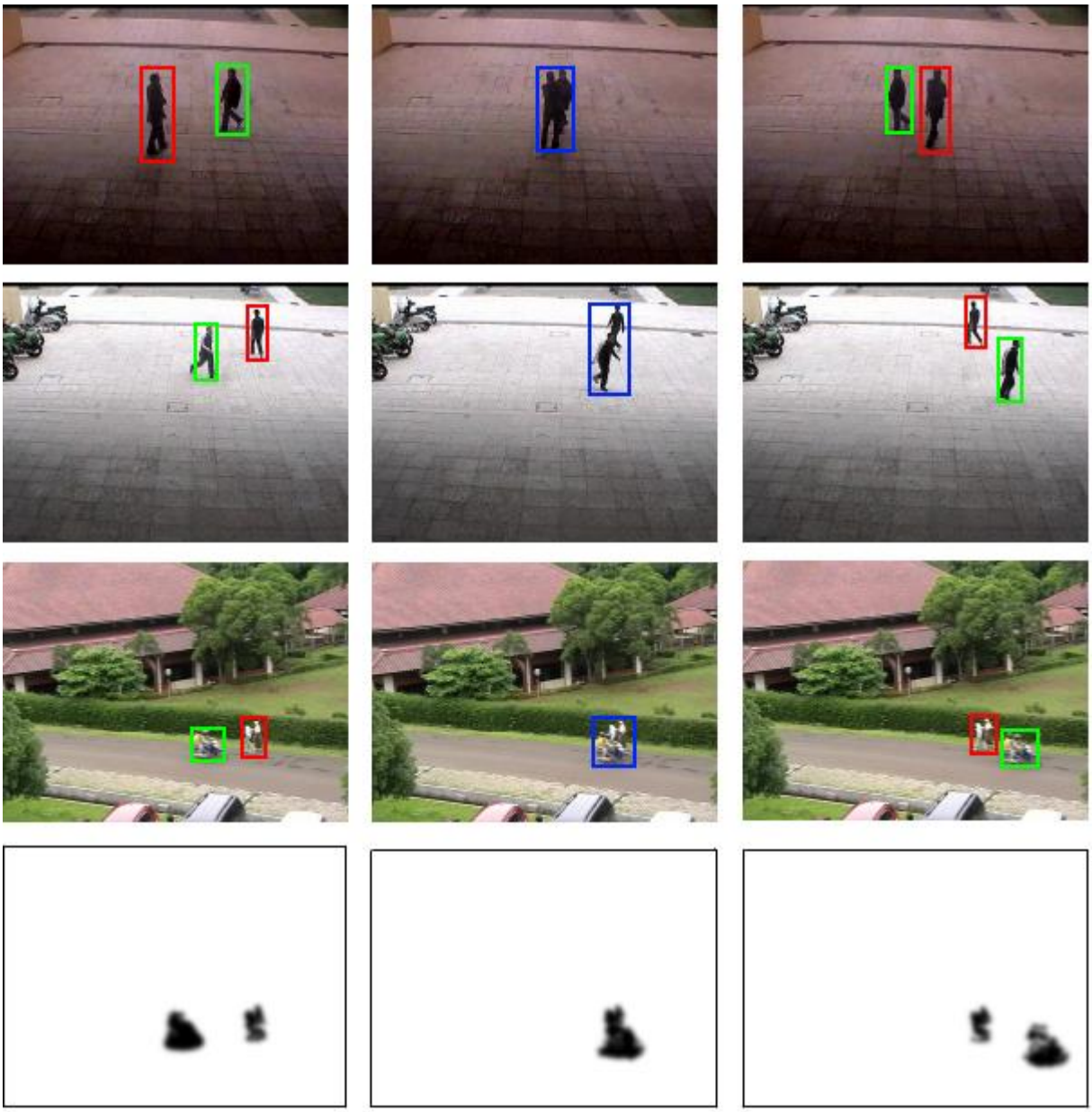

Fig. 7. Sample tracking results and occlusion resolution by using the proposed method. In each row, the objects are shown before, at, and after occlusion. The fourth row shows the corresponding blob result for the images in the third row.

## 6. EXPERIMENT RESULTS

The experiment is conducted to show the effectiveness of the proposed method to handle occlusion in real world video scenes. The data used in the experiment are obtained from real traffics at Universitas Indonesia campus and other video benchmark available in the Internet[1]. The objects in video can be pedestrians, cars, or bikes.

Figure 7 shows the result of the proposed method on the video dataset. In each row, there are two objects in the video each of which is marked by red and green rectangle. When the two objects occluded, merging event is detected and the occlusion is handled by creating a new track, marked by blue box, regarding the newly formed group. The tracks of individual objects are frozen during occlusion. When the group splits, these tracks are recalled. The similarity measure is performed to restore the tracks as it was before occlusion occurred.

In the first row, the objects seem similar to each other. In this situation, a slight distortion of the features could cause

[1] CAVIAR (http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/), PETS 2001 (http://ftp.pets.rdg.ac.uk/PETS2001/), VISOR (http://imagelab.ing.unimore.it/visor/video_categories.asp)

false similarity comparison. By using multiple features, this problem is addressed. Namely, when similarity calculation on a particular feature gives false result other features could correct the error. This result is consistent with the conclusion in [7] that using multiple features would increase the tracking accuracy.

Illumination change appears quite frequently in real video scenes. The second row of Figure 7 represents the situation when two objects occupied different illumination area before and after occlusion. The error caused by this problem is reduced by adjusting the feature values according to the current illumination. Incorporating colorspace that robust against illumination differences would also be helpful. In the proposed method, hue and value component of the HSV colorspace are used. The hue component indicates the major color regardless the brightness of the image. Assigning greater weight to the hue component would increase the feature robustness against illumination change. The value component is useful when there is nearly no colors present in the image [15] or the hue components are similar.

The fourth row of the image shows the corresponding extracted blobs of the objects in the third row. As shown in the figure, the extracted blobs are somewhat not accurate. In practice, it is difficult to obtain very accurate blob results in all frames. Background subtraction could perform poorly if the foreground is similar to the background or in presence of fast illumination change. Moreover, the method depends on many parameters to fit in certain conditions. Thus, the challenge to the tracking system is to deal with this limitation. Given this condition, the proposed method can still perform similarity measure effectively and accurately resolve the tracks. It shows the good performance of fuzzy similarity measure in dealing with uncertainty and imprecision. The fuzzy approach provides a fault tolerance solution to the problem. The update procedure is also helpful since it can avoid instability of feature extraction caused by inaccurate blob extraction in certain frames.

## 7. Computational Time Analysis

Running time of the proposed method can be analyzed by dividing the overall process into its components. The major processes are: background subtraction, connected component analysis, object correspondence, feature extraction and model updating, and occlusion handling. These processes run for each retrieved frame in sequential manner.

For background subtraction, Adaptive GMM technique requires O($m$) time for each pixel [7], where $m$ is the number of Gaussian distribution. Connected component analysis is performed by using linear time algorithm proposed in [15], so the required time is O($h$×$w$), where $h$ and $w$ are height and weight of the video frame and $h$×$w$ equals to the total number of pixels in a frame. In object correspondence, each pair of blob and tracks are examined, so the total time is O($k^2$) where $k$ is maximum number of objects. For feature extraction and model updating, the running time is determined by the number of partitions in fuzzy histograms for color ($b_h$, $b_v$) and texture ($b_t$) multiplied by the maximum number of objects $k$ and the maximum blob size $s$. Whilst for occlusion handling, the process is dominated by similarity measure for occlusion resolution. The complexity is determined by the number of partitions for fuzzy histograms for color $b_h$, $b_v$ and texture $b_t$. The overall complexity is the sum of all components which is ($h$×$w$×$m$) + ($h$×$w$) + ($k^2$) + ($s$×$k$×($b_h$+$b_v$+$b_t$)) + ($k$×($b_h$+$b_v$+$b_t$)).

In the experiment, the system is run in 2 GHz Intel Pentium Dual CPU with 2 GB of RAM. The average processing time per frame was 94.5 milliseconds, so the frame processing rate is ≈ 10. This rate is fast enough for CCTV video surveillance which do not employ high frame rate for storage efficiency reason.

## 8. Conclusions

Occlusion is a common and serious problem in multi-object tracking. It can degenerate the tracking accuracy if not properly handled. In this paper, a method for resolving occlusion is proposed by using fuzzy approach. The occlusion handling procedure consists of two parts: merging-splitting detection and resolving object tracks after occlusion. Merging event is detected by using object bounding box while splitting event is detected by searching any group around. When a group splits, the tracks of the objects are resolved by performing fuzzy similarity measures to find its corresponding track before occlusion happened. Fuzzy similarity measure is incorporated in three steps: (1) calculating the feature of the object in fuzzy representation, (2) calculating similarities for each feature, and (3) aggregating the individual similarities to obtain the total similarity value. The experimental results showed the effectiveness of the proposed method and its robustness against inaccurate foreground extraction, similarity between objects, and illumination differences. The results also showed that the method runs in moderately fast time and hence, is suitable for real video surveillance application.

## References

[1] Hu, W., Tan, T., Wang, L., Maybank, S. 2004. A Survey on Visual Surveillance of Object Motion and Behaviors. IEEE Trans On System, Man and Cybernetics – Part C: Applications and Reviews, 34(3).

[2] Gabriel, P. F., Verly, J.G., Piater J. H., Genon, A. 2003. The State of the Art in Multiple Object Tracking Under Occlusion in Video Sequences. Proc. of Adv. Concepts for Intelligent Vision Syst. (ACIVS, Sep. 2003): 166-173.

[3] McKenna, S. J., Jabri, S., Duric Z., Wechsler, H., Rosenfeld, A. (2000). Tracking Groups of People. Computer Vision and Image Understanding, 80(1): 42-56.

[4] Javed, O., Shah, M. 2002. Tracking and Object Classification for Automated Surveillance. LNCS; Proc. of the 7th European Conf. on Computer Vision-Part IV, 2353: 343-357, Springer-Verlag.

[5] Yang, T., Li, S. Z., Pan, Q., Li, J. 2005. Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes. Proc. CVPR'05 – 1: 970-975.

[6] Senior, A., Hampapur, A., Tian, Y. L. (2006). Appearance Models for Occlusion Handling. Image and Vision Computing, 24(11): 1233-1243.

[7] Takala, V., Pietikäinen, M. (2007). Multi-Object Tracking Using Color, Texture, and Motion. CVPR 2007, 18-23 June 2007.

[8] Swain, M. J. & Ballard, D. H. (1991). Color Indexing. International Journal of Computer Vision, 7(1), November 1991. Springer Netherlands.

[9] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR 1997), San Juan, Puerto Rico, 17-19 June 1997, pages 762–768, 1997.

[10] Mäenpää, T. & Pietikäinen, M. 2005. Texture Analysis with Local Binary Pattern. Ch 1, in C. Chen and P. Wang (eds) Handbook of Pattern Recognition and Computer Vision, 3rd ed: 197-216. World Scientific.

[11] Tizhoosh, H. R. (June 1997, Updated November 2004). Why Fuzzy Image Processing?. Accessed 19 Jan 2010 website: http://pami.uwaterloo.ca/tizhoosh/why.htm.

[12] Piccardi, M. (2004). Background Subtraction Techniques: a Review. IEEE Intl. Conf. on Systems, Man and Cybernetics (Oct 2004), 4: 3099-3104.

[13] Stauffer, C. & Grimson, W. E. L. 2002. Adaptive Background Mixture Models for Real-Time Tracking. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (2), 1999.

[14] KaewTraKulPong P. & Bowden, R. 2001. An improved adaptive background mixture model for real-time tracking with shadow detection. Proc. 2nd EU Workshop on Adv. Video-Based Surveillance Syst, Sep 2001.

[15] Nachtegael, M., Schulte, S., De Witte, V., Mélange, T., Kerre, E.E. 2007. Image Similarity – From Fuzzy Sets to Color Image Applications. LNCS 4781, pp.26-37, 2007.

[16] Mardiko, R. & Widyanto, M. R. 2009. An Approach to Object Counting for Video Surveillance Using Fuzzy Inference System with Fault Tolerance. Proc. Intl Conf on IT Application and Management, UI Depok, Apr 2009.

[17] Chamorro-Martínez, J., Sánchez, D., Soto-Hidalgo, J.M. (2008). A Novel Histogram Definition for Fuzzy Color Spaces. IEEE Intl. Conf. on Fuzzy Systems 2008.

[18] Mardiko, R. Fuzzy Similariy Measures in CBIR. Technical report, Faculty of Computer Science Universitas Indonesia.

[19] Santini, S. & Jain, R. 1999. Similarity Measures. IEEE Trans. *PAMI*, 21(9): 871-883, Sep 1999.

[20] Omhover J. F., Detyniecki, M., Bouchon-Meunier, B. 2004. A Region-Based Image Retrieval System. Proc. of IPMU'04, Perugia, Italy, July 2004.

[21] Grabisch, M. (1996). The Application of Fuzzy Integrals in Multicriteria Decision Making. European Journal of Operational Research, 89:445-456.

[22] Chen, Y. X., Wang, J. Z. 2002 . A Region-Based Fuzzy Feature Matching Approach for Content-Based Image Retrieval. IEEE Trans. PAMI, Vol 24 No. 9, September 2002

[23] Jiang, W., Er, G., Dai, Q., Gu, J. 2006. Similarity Based Online Feature Selection in Content Based Image Retrieval. IEEE Trans. on Image Processing, Vol. 15(3), March 2006.

[24] Chang, F. 2004. A linear-time component-labeling algorithm using contour tracing technique. Computer Vision and Image Understanding, 93(2):206-220.