Multiple Object Tracking under Occlusion Conditions

Young-Kee Jung[a] and Yo-Sung Ho[b]

[a]Honam University
59-1 Seobong-dong Kwangsan-gu, Kwangju, 506-090, Korea

[b]Kwangju Institute of Science and Technology
1 Oryong-dong Puk-gu, Kwangju, 500-712, Korea

ABSTRACT

This paper describes an algorithm for multiple object tracking that takes a new occlusion reasoning approach. In order to track individual objects under occlusion conditions, we design a two-dimensional token-based tracking system using Kalman filtering. The proposed tracking system consists of two parts: object detection and tracking, and occlusion reasoning using feature matching. The object detection and tracking part finds moving objects from their background. For object detection, we develop an adaptive background update technique. By tracking individual objects with segmentation information, we generate motion trajectories. Computer simulation of the proposed scheme demonstrates its robustness to various occlusion conditions for several test sequences.

Keywords: Object Detection, Object Tracking, Occlusion Reasoning, Feature Matching, Visual Surveillance

1. INTRODUCTION

In video surveillance applications, object tracking algorithms are employed to track detected objects and predict their locations based on a linear predictor model. For multiple object tracking, it is important to continue to track those objects even if other moving objects in the scene occlude with them. In the case of traffic monitoring, vehicles overlapping each other within the image plane have caused problems of measurement accuracy.

An occlusion reasoning algorithm is proposed by Malik [1]. The algorithm analyzes moving vehicles in the order of their depth (distance to the camera) and preserves the shape mask within the overlapping region. However, this approach does not resolve the initial occlusion problem where vehicles appear too close to one another in the scene. This approach is appropriate only for the contour-based tracking system. Other reasoning algorithms using occlusion prediction are proposed to detect a collision of two or multiple object by estimating bounding boxes of objects and to predict the next position until the end of occlusion using the old parameters [2]. However, this approach is likely to have a problem of correspondence mismatch when the behavior of the object is nonlinear or the prediction parameters before the occlusion have some errors. This approach does not consider the initial occlusion problem, either.

In order to solve the correspondence mismatch and the initial occlusion problems, we propose two different types of occlusion reasoning algorithms: explicit occlusion and implicit occlusion. Explicit occlusion represents the following situation: after two or more objects appear separately, they are merged due to some occlusion conditions. For the case of explicit occlusion, we suggest a feature matching technique to reduce the possibility of the correspondence mismatch. We utilize various features that are used to match objects of pre-occlusion to objects of post-occlusion. Implicit occlusion represents the initial occlusion where two or multiple objects appear as a single object in the scene.

2. OBJECT DETECTION AND TRACKING

In order to detect moving objects from a dynamic background scene, we develop an adaptive background update method, where we do not update the background of the moving objects. For the adaptive background update, we utilize the temporal median operation [3,4].

[b]Correspondence: E-mail:hoyo@kjist.ac.kr; Telephone:+82-62-970-2211; Fax:+82-62-970-2204

Fig. 1 depicts our proposed scheme for moving object detection.

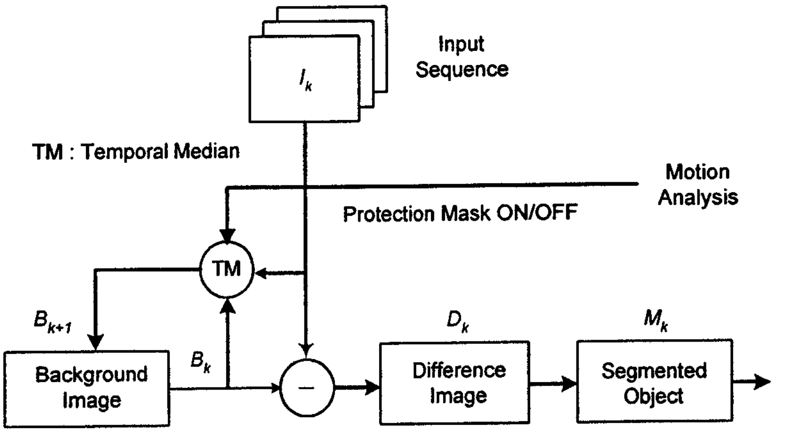


Figure 1. Proposed Scheme for Moving Object Detection

For frame I_k in the video sequence, the background B_k is updated as Eq. (1).

$$B_{k+1}(x) = \begin{cases} B_k(x)+1, & \text{if } B_k(x) < I_k(x) \\ B_k(x)-1, & \text{otherwise} \end{cases} \tag{1}$$

Over a period of time, the background image contains the temporal median values of pixels for the image position x. The moving object mask M_k is obtained by thresholding the absolute difference of the input image I_k and its background B_k at discrete time k.

Since we are interested in the motion of moving objects for visual surveillance, we should be able to distinguish between the noisy motion of temporal clutters and the actual motion of objects. For this purpose, we devise a motion classification rule. When there are N moving objects at time k, we represent all objects by VO_k^p $(p=1,...N)$. Each object has a set of features, such as the shape mask, the center position and the rectangular bounding box. An example of object tracking is illustrated in Fig. 2, where there are two motion trajectories. The first motion trajectory has six tracking positions and the second motion trajectory has five tracking positions.

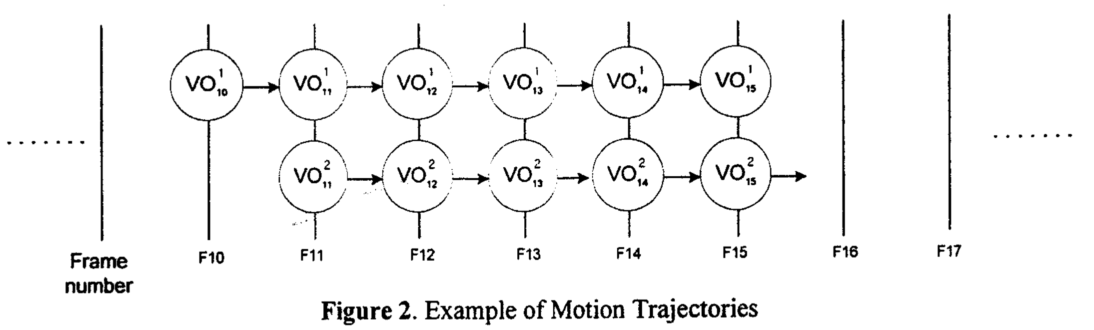


Figure 2. Example of Motion Trajectories

Three different patterns of motion are defined: (i) *rest motion*, representing the motion of an object which enters into the scene and stops, (ii) *moving motion*, representing a moving object in the scene, and (iii) *stop motion* of temporal clutter, which is the noisy motion of no importance for visual surveillance, such as swaying plants.

Since the noisy motion usually occupies a limited range of variations during its lifetime, we design a motion classification rule, which is described in Fig. 3. The rule-based motion classification algorithm enables or disables the protection mask according to the object motion during its lifetime. In case of *stop motion* of a temporal clutter, the noisy object is removed during the background image update. Therefore, the noisy object motion trajectory has a short lifetime and a limited moving range. In case of *rest motion* of the actual object, the object has significant movement during the initial period of its life span. Therefore, the proposed rule generates a protection mask of the object region and preserves the rest object.

```
Algorithm for Rule-based Motion Classification
Begin
    For the life span of each motion trajectory
        if all tracking positions are clustered in a certain area
            motion trajectory of VO  = STOP;
            disable the protection mask;
        else
            motion trajectory of VO  = REST or MOVING;
            enable the protection mask;
    If motion trajectory of VO == STOP
        remove trajectory of VO;
END
```

Figure 3. Motion Classification Rule

Once we detect a moving object, we can track the moving object efficiently by predicting the next center coordinate from the observed coordinate of the moving object. We design a token-based tracking scheme using Kalman filtering [5]. The center position and the size of the moving object are used as the token $t(k)$.

We represent the next token $t(k+1)$ as the sum of the current token $t(k)$ and the token change $\Delta t(k)$. We can define a simplified polynomial motion model as follows:

$$t(k+1) = t(k) + \Delta t(k) \tag{2}$$

Kalman filtering provides a sequential and recursive algorithm for optimal linear minimum variance (LMV) estimation of the system state $x(k)$. We define the state variable as a four-dimensional vector, which represents the positional change of the target object per unit time interval and the size change of the target object.

$$x(k) = \begin{pmatrix} \Delta x\_center(k) \\ \Delta y\_center(k) \\ \Delta xsize(k) \\ \Delta ysize(k) \end{pmatrix} \tag{3}$$

We assume that the state model is linear and defined by

$$x(k+1) = \Phi(k,k+1)x(k) + w(k) \tag{4}$$

$$\Phi(k,k+1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

where $x(k)$ denotes the system state at time instant k, $\Phi(k,k+1)$ is a state transition matrix during the unit time interval, and $w(k)$ represents an estimation error. Assuming that the trajectory of the target object varies with a constant acceleration and the size of the target object varies linearly, we can write the state transition matrix as in Eq. (5). We can also assume a linear relationship between the system state and a set of measurements.

$$z(k) = H(k)x(k) + v(k) \tag{6}$$

$$H(k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

where $x(k)$ denotes a set of measurements, $H(k)$ is an observation matrix, and $v(k)$ represents measurement errors.
After we define the system model and the measurement model, we can apply a recursive Kalman filtering algorithm to obtain LMV estimates of motion parameters. The recursive Kalman filtering algorithm consists of three steps: initialization, state prediction, and measurement update. The initialization step determines the initial state estimate and the initial error covariance matrix which represents deviation of the initial state estimate from the actual initial state. After the initialization step, we switch to the tracking mode. In the state prediction step, we determine a priori LMV estimate and its error covariance matrix for the current state, based on the previous state estimate and error covariance. In the measurement update step, we combine the estimated information with new measurements in order to refine the LMV estimate and its error covariance matrix for the current state. We perform this correction process with measurements on the positional change and the size of the target object.

3. OCCLUSION REASONING USING FEATURE MATCHING

For multiple object tracking, the tracking algorithm needs to make data association in order to decide which of the observed objects should be associated with which track. This data association problem can be solved by a nearest neighbor prediction approach, which associates the nearest neighbor object within the prediction region to each track [6]. However, in some occlusion situation shown in Fig. 4, the nearest neighbor prediction method does not provide correct data association.

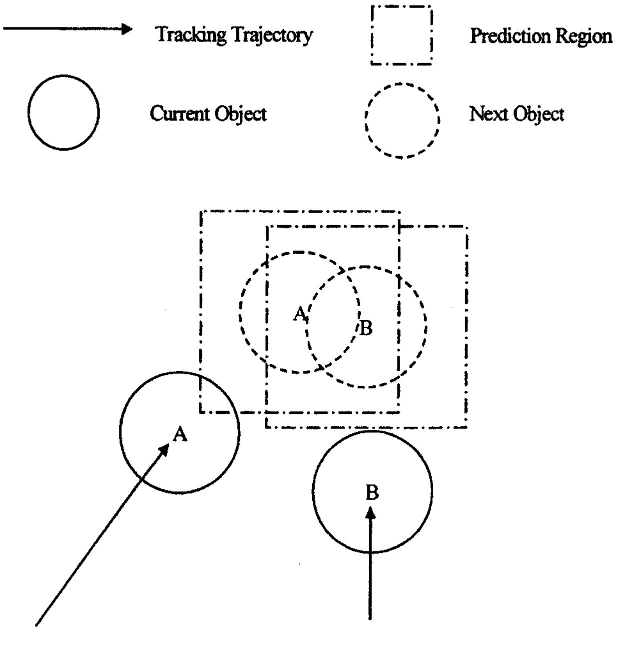


Figure 4. Data Association in Occlusion Condition

In order to solve this occlusion problem, several occlusion reasoning algorithms try to detect the occlusion region and apply a nearest neighbor association rule after the separation. As shown in Fig. 5, once a collision of object A and object B is detected in the predicted region, the prediction of the next position is in progress until the end of occlusion using the old parameters. If the occluded objects are separated, the correspondence could be solved using the nearest neighbor rule. However, if the prediction error is accumulated during the occlusion, this may cause a correspondence mismatch problem, as shown in Fig. 6.

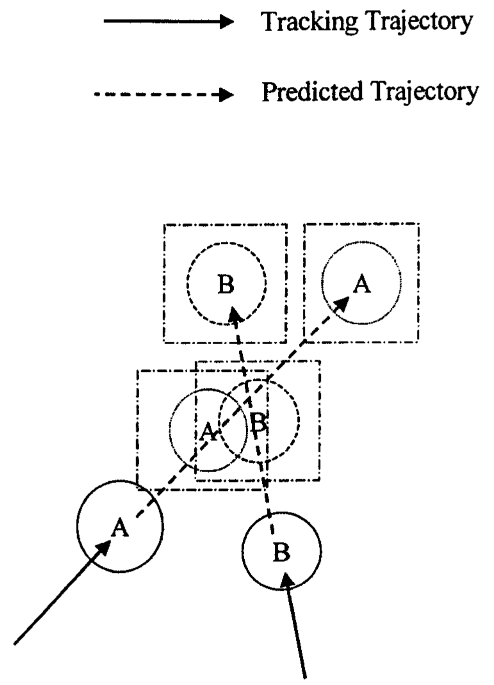


Figure 5. Previous Occlusion Reasoning Algorithms

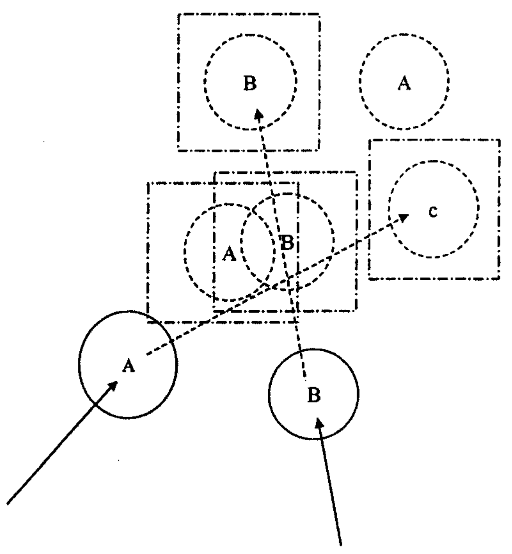
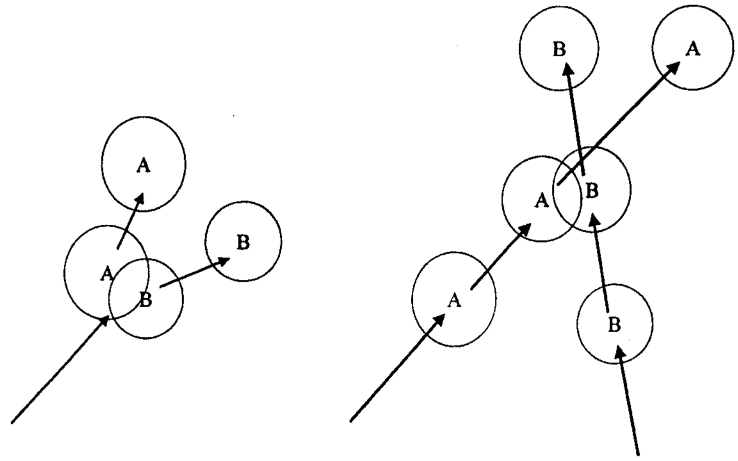


Figure 6. Problem of Correspondence Mismatch

In this paper, we propose a new occlusion reasoning algorithm. In Fig. 7, we show two different types of occlusions: explicit occlusion and implicit occlusion.



(a) Implicit Occlusion (b) Explicit Occlusion

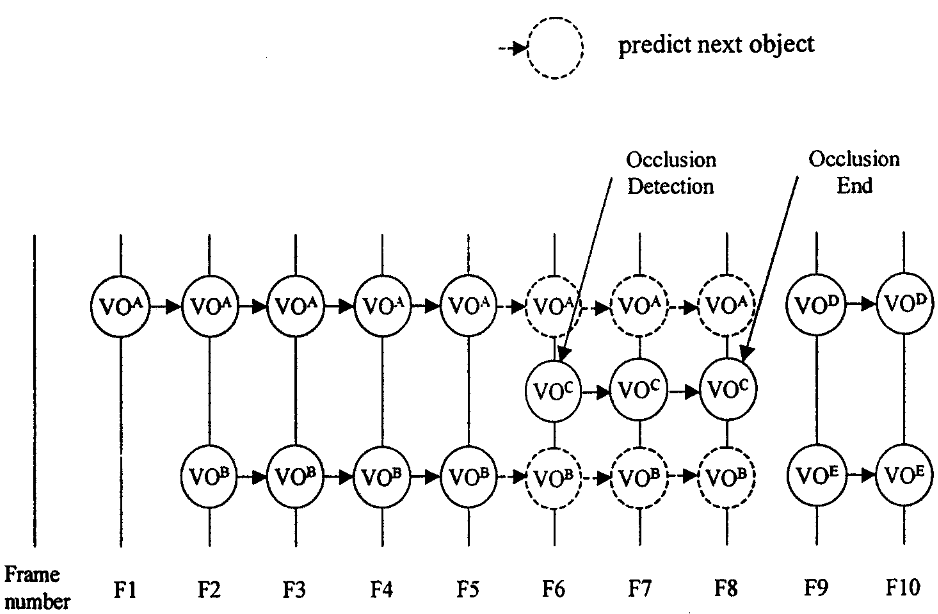
Figure 7. Two Types of Occlusions

Each node of the motion trajectory is defined by

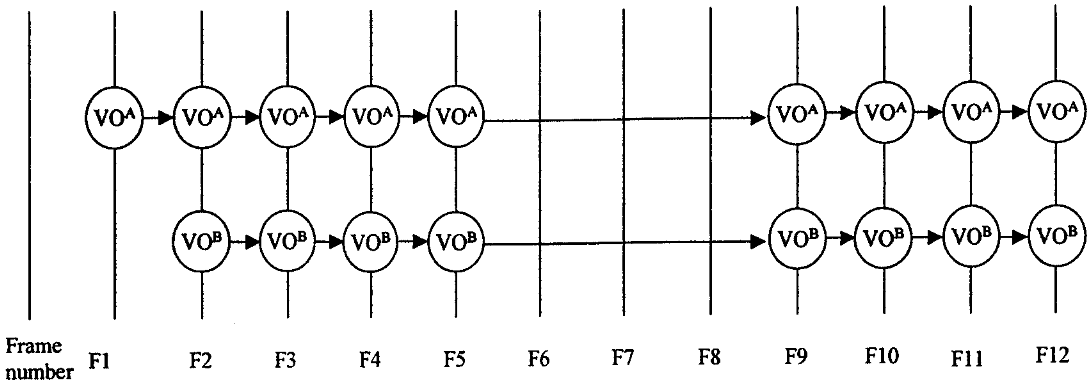
$$VO_k^p(p=1,...,N)=f(M,C,R,V) \tag{8}$$

where k denotes the discrete time, N is the number of moving objects, $f(\cdot)$ is a function of a set of features, including the shape mask(M), the center position(C), the bounding rectangle(R), and the velocity(V).

When the occluded objects are separated, we can check whether the occlusion type is explicit or implicit. If the occluded objects have a collision, the occlusion type is explicit. In the case of the explicit occlusion, we apply a feature matching algorithm to make correspondence between the objects before and after the occlusion, as shown in Fig. 8.



(a) Merging and Separation

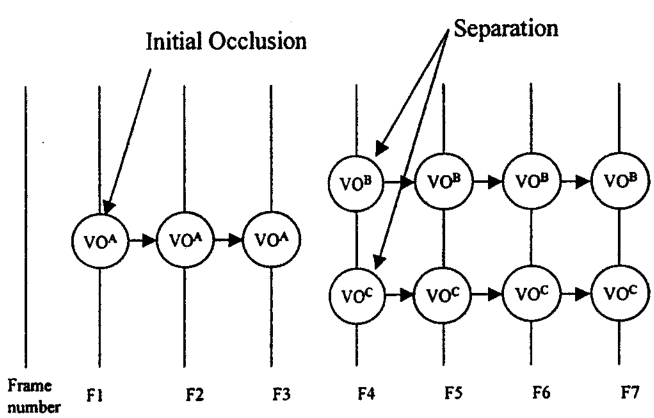


(b) Correspondence by Feature Matching

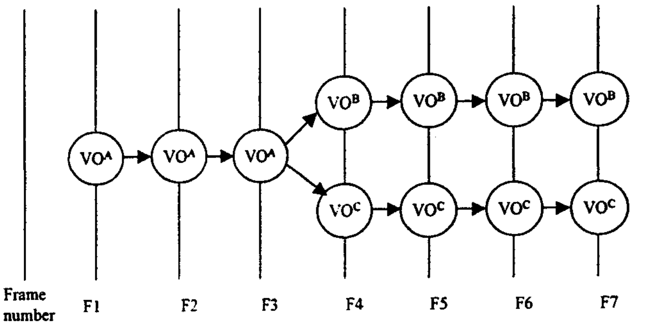
Figure 8. Explicit Occlusion Reasoning

- **Explicit Occlusion Reasoning (Fig. 8)**
  (a) The tracking algorithm detects a collision of two or multiple objects using the estimated bounding boxes of objects.
  (b) Once a collision of objects A and B is detected, the reasoning algorithm creates a new trajectory of the merged object C.
  (c) The reasoning algorithm tracks the merged object C continuously until the end of occlusion, and predicts the objects A and B simultaneously using the parameters before occlusion.
  (d) When the merged objects are separated as objects D and E, we associate the nearest neighbor object with each track within the prediction region. In order to validate the data association by feature matching, we use Eq.(9) to select two trajectories that have the minimum feature distance.

$$\begin{aligned}
&\textit{for each } i \\
&\quad \underset{j}{\operatorname{argmin}} M_{i,j} = \sum_{k} w_k \left| f_i(k) - f_j(k) \right| \\
&i = A, B \\
&j = D, E \\
&M_{i,j} = \text{mesure for matching} \\
&\qquad \text{between } i \text{ and } j \text{ trajectories} \\
&w_k = \text{weighting factor} \\
&f(k) = \{\text{velocity, size, gray intensity}\}
\end{aligned} \tag{9}$$



(a) Separation



(b) Connection

Figure 9. Implicit Occlusion Reasoning

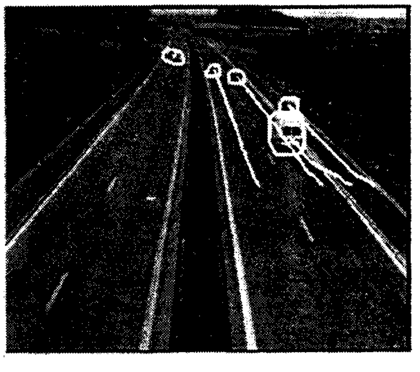
In the case of implicit occlusion, the initial occlusion reasoning algorithm is used.

- **Implicit occlusion reasoning (Fig. 9)**
  (a) If the merged object A is separated, the reasoning algorithm creates two trajectories of the separated objects B and C.
  (b) The reasoning algorithm connects the common trajectory A to trajectories of B and C.

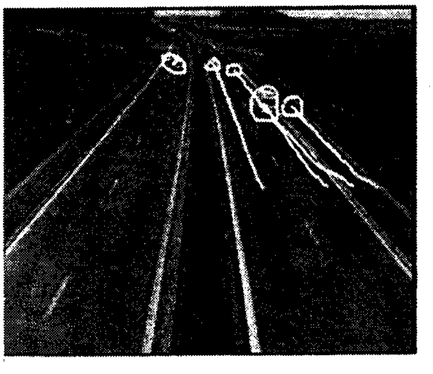
4. EXPERIMENTAL RESULTS

Computer simulations have been performed on two video sequences of highway traffic scenes and one road sequence. The highway traffic sequences and the road sequence belong to the MPEG-7 video test materials [7] and each sequence has the ITU-R 601 format. The highway sequences contain two traffic lanes and one emergency lane for each traveling direction. The sequences are taken from the same camera and contain some abnormal behavior of vehicles, which travel emergency lane and overlap with other vehicles. The road sequence contains various types of motion patterns, such as walking men, traveling of vehicles and bicycles.

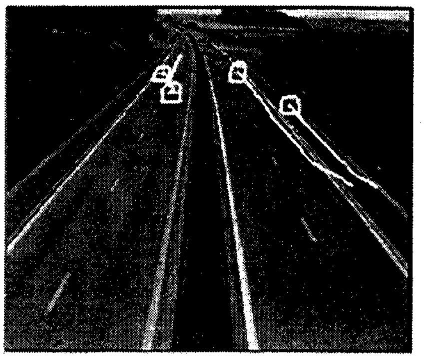
We evaluate the proposed occlusion reasoning algorithm for explicit and implicit occlusion situations. In Fig. 10(a) and Fig. 10(b), one vehicle enters the scene on the right emergency lane, stops and occludes with the other vehicle. Therefore, the occlusion type is explicit. The explicit algorithm predicts two objects using the parameters before occlusion and creates a new trajectory of the merged object. When the merged objects are separated, the nearest neighbor object within each prediction region is associated with each track and is validated by feature matching. The tracking trajectory of the right-side road in Fig. 10(b) is the result that we obtain by the explicit occlusion reasoning algorithm. In this case, because two vehicles have significant difference of speeds and gray intensity values, the feature matching algorithm easily validates the correspondence.



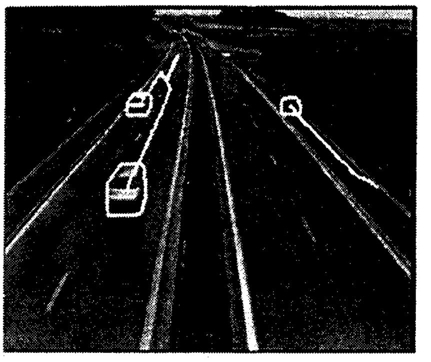
(a) Frame # 3526



(b) Frame # 3539



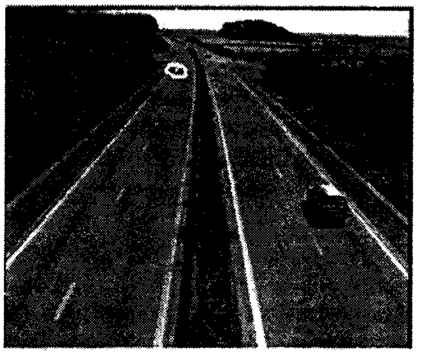
(c) Frame # 3569



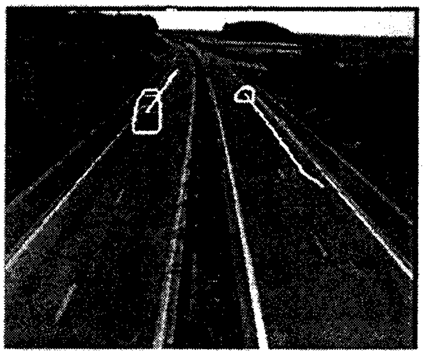
(d) Frame # 3598

Figure 10. Explicit/Implicit Occlusion Reasoning in the Highway Traffic Sequence

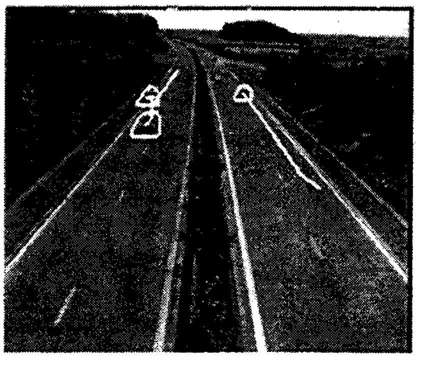
Since two vehicles in Fig. 10(a) appear too close to each other in the left side of the road, it is difficult to separate the two vehicles by the object segmentation algorithm. In Fig. 11(a), one vehicle passes another vehicle and drives towards the camera, creating a visual occlusion. For these two situations of initial occlusions, we apply the implicit reasoning algorithm to generate two tracking trajectories. If the merged object is separated, the reasoning algorithm creates two trajectories of the separated objects. Then, the reasoning algorithm connects the old trajectory to the new trajectories, as shown in Fig. 10(c) and Fig. 11(c).



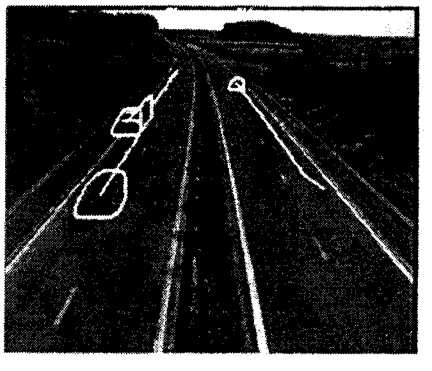
(a) Frame # 1765



(b) Frame # 1823

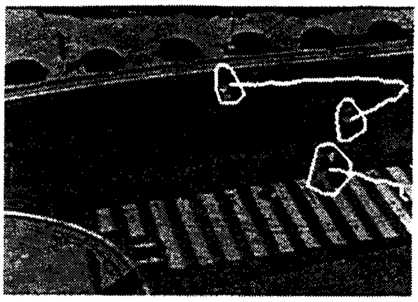


(c) Frame # 1824

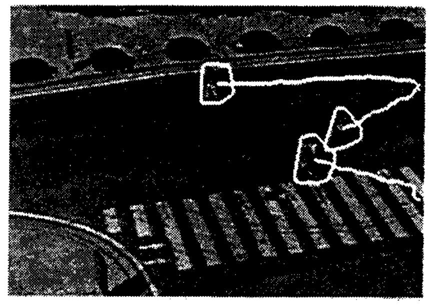


(d) Frame # 1841

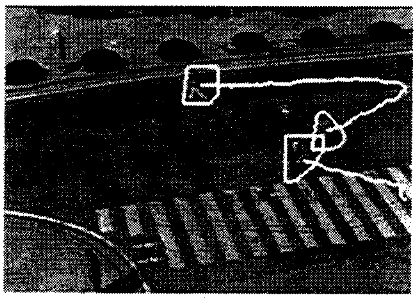
Figure 11. Implicit Occlusion Reasoning in the Highway Traffic Sequence



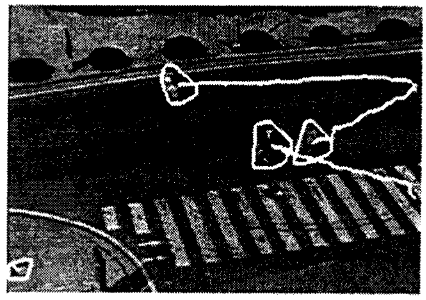
(a) Frame # 1640



(b) Frame # 1660



(c) Frame # 1666



(d) Frame # 1690

Figure 12. Explicit Occlusion Reasoning in the Road Sequence

Fig. 12 shows experimental results of the explicit occlusion reasoning on the road sequence. As shown in Fig. 12(a) and Fig. 12(b), the explicit occlusion reasoning algorithm detects the partial occlusion of two objects which is caused by the shadow cast of one object. In this case, the speed and the size of two objects are similar. Since two objects have different gray-level intensity values, the gray-level intensity value becomes a major feature to validate the correspondence.

5. CONCLUSION

In this paper, we propose a new scheme for multiple object tracking under occlusion conditions for visual surveillance applications. In order to extract a motion trajectory of the moving object, we devise novel algorithms for object detection and tracking. The proposed algorithm for object detection removes noisy objects efficiently by motion classification and prohibits the background image from assimilating the object of a stationary motion. The proposed occlusion reasoning algorithm works quite well for occlusion situations and reduces the possibility of the correspondence mismatch due to trajectory matching using three features, i.e., the size, the speed, and the gray level intensity values of the object.

ACKNOWLEDGEMENTS

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST), and in part by Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

REFERENCES

1. J. Malik and S. Russell, "A Machine Vision Based Surveillance System for California Roads", PATH Project MOU- 83 Final Report, University of California, Berkeley, 1994.
2. R. Rosales and S. Sclaroff, "Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling", IEEE Conf. On Computer Vision and Pattern Recognition, pp.127-134, June 1998.
3. N. McFalane and C. Scholfield, "Segmentation and Tracking of Piglets in Images", Machine Vision and Application, Vol. 8, pp. 187-193, October 1995.

4. Y.K. Jung and Y.S. Ho, "Robust Vehicle Detection and Tracking for Traffic Surveillance", Picture Coding Symposium'99, pp. 227-230, April 1999.
5. A. M. Tekalp, Digital Video Processing, Prentice Hall, 1993.
6. B. Rao, H. Durrant-Whyte and J. Sheen, "A Fully Decentralized Multi-Sensor System For Tracking and Surveillance", The International Journal of Robotics Research, Vol. 12, No. 1, pp.20-44, February 1993.
7. Requirements Group, "Licensing Agreement for the MPEG-7 Content Set," ISO/IEC JTC1/SC29/WG11/MPEG98/ N2466, Atlantic City, October 1998.