第 29 卷 第 3 期
2003 年 5 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 29, No. 3
May, 2003

Tracking Occluded Objects Using Partial Observation[1)]

Ming Xu Tim Ellis
(*Information Engineering Centre, City University, London, EC1V 0HB, U. K.*)
(E-mail: {m. xu, t. j. ellis}@city. ac. uk)

Abstract This paper presents a framework for multi-object tracking from a single fixed camera. The potential objects to track are detected with intensity-plus-chromaticity mixture models. The region-based representations of each object are tracked and predicted using a Kalman filter. A scene model is created to help predict and interpret the occluded or exiting objects. Unlike the traditional blind tracking during occlusion, the object states are estimated using partial observations whenever available. The observability of each object depends on the predictive measurement of the object, the foreground region measurement, and the scene model. This makes the algorithm more robust in terms of both qualitative and quantitative criteria.

Key words Partial observation, scene model, foreground region

1 Introduction

Tracking non-rigid targets over large ranges of depth has long been realized as a region-based correspondence problem, in which each target is mapped from one frame to the next according to its position, dimension, colour and other contextual information. When multiple targets exist and their dimensions are not negligible in comparison with their velocities, occlusion or grouping of these targets is a routine event. This brings about uncertainty for the tracking, because the contextual information is only available for the group and individual targets cannot be identified.

Existing region-based tracking algorithms use either the measurement for the group or the prediction for each target to update the target estimate through grouping. Intille, Davis and Bobick updated the centroid of a target using the group measurement and held the velocity, size and colour estimates[1]. Rosales and Sclaroff modelled the two corners of each target's bounding box and updated their positions using the prediction of an Extended Kalman filter[2]. Ellis and Xu estimated the target, which is closer to the group in state distance, using the group measurement and updated the other targets with prediction[3]. However, these algorithms all suffer from poor performance for target estimation during grouping or occlusion. To estimate a target with the group measurement, the estimate of the target is often seriously discontinuous at the start of grouping and may be so misleading as to fail to find a match at the end of grouping. Target updating using prediction is heavily reliant on the motion model and vulnerable to any violation of the underlying assumption during grouping, e. g. the target turning or accelerating, for a first-order motion model that assumes a linear trajectory and a constant velocity.

We realize that the targets in a group are often partially observable, because some of their bounding edges constitute the four bounding edges of the group. If these partial observations are fed into the estimation process during grouping, the tracker should be more robust and accurate than those without any observation[4]. In this paper, our system assumes that each target has a constant height and width, and models the four bounding edges of each target using a Kalman filter. Once some edge is decided to be observable and its measurement is input to the tracker, its opposite, unobservable edge could be roughly deduced because the two opposite edges share the "same" (though disturbed by noise) hori-

1) Supported by the EPSRC(GR/M58030)
Received November 28, 2002; in revised form April 7, 2003
收稿日期 2002-11-28; 收修改稿日期 2003-04-07

zontal or vertical velocity according to the constant height and width assumption. The deduction of unobservable variables from observable ones can be either direct or implicit. The decision of target observability is based on the group foreground measurement, target predictive measurement and a simple scene model.

A related work[5] models a pair of hands by two bounding boxes and tracks them using the CONDENSATION algorithm. It considers an edge to be erroneously observed, when the observation differs from prediction by a large value, and estimates the "correct" observation. Our algorithm is different from [5] in that it copes with an unlimited number of targets and the observability for a target is decided not only by the underlying target but also by the others in the same group. In addition, we weight the uncertainty of the estimated "observation" and account for static occlusions. Another related work[6] tracks the bounding edges of each target using a Kalman filter and weights the measurement covariance with the "visibility" of the bounding edges. It uses optical flow analysis to segment the observations for individual targets in grouping, which further decides the "visibility" in tracking. Our algorithm is more efficient in that it does not need optical flow analysis.

This paper is organized as follows: in Sections 2 and 3 the foreground detection and the Kalman filter are described; in Section 4 a scene model for static occlusions is introduced and followed by the concept of partial observation and tracking algorithm in Section 5; experimental results are illustrated in Section 6.

2 Foreground measurement

Our system uses frame differencing for change detection in dynamic images. It compares each incoming frame with an adaptive background image and classifies those pixels of significant variation into foreground. To maintain a reliable, illumination-invariant change detector, the probability of observing values, $\boldsymbol{y}_{k,I}(\boldsymbol{x}) = I = R + G + B$ and $\boldsymbol{y}_{k,c}(\boldsymbol{x}) = [R/I, G/I]$, at pixel $\boldsymbol{x}$ is modelled by two mixtures of Gaussians[7,8]. The i-th Gaussian distribution in each mixture model at time k is characterised by its (temporal) mean $\boldsymbol{\mu}_{i,k}(\boldsymbol{x})$, trace of the covariance matrix, $\sigma^2_{i,k}(\boldsymbol{x})$, and weight, $\omega_{i,k}(\boldsymbol{x})$, reflecting the likelihood that the i-th distribution accounts for the data.

At time k, every new pixel value is checked against the Gaussian distributions in a mixture model. For a matched distribution i, the pixel measurement is incorporated in the estimate of that distribution and the weight is increased:

$$\begin{aligned} \boldsymbol{\mu}_{i,k}(\boldsymbol{x}) &= (1-\beta)\boldsymbol{\mu}_{i,k-1}(\boldsymbol{x}) + \beta \boldsymbol{y}_k(\boldsymbol{x}) \\ \sigma^2_{i,k}(\boldsymbol{x}) &= (1-\beta)\sigma^2_{i,k-1}(\boldsymbol{x}) + \beta \| \boldsymbol{y}_k(\boldsymbol{x}) - \boldsymbol{\mu}_{i,k}(\boldsymbol{x}) \|^2 \end{aligned} \tag{1}$$

where β controls the background updating rate and $\beta \in (0,1)$. For unmatched distributions, their estimates remain the same but the weights are decreased. If none of the existing distributions matches the current pixel value, either a new distribution is created, or the least probable distribution for the background is replaced. The distribution with the greatest weight, i_B, is identified as the a priori background model for time $k+1$. At time k, the set of foreground pixels identified is:

$$F_k = \{\boldsymbol{x} : \| \boldsymbol{y}_k(\boldsymbol{x}) - \boldsymbol{\mu}_{i_B,k-1}(\boldsymbol{x}) \| > 2.5\sigma_{i_B,k-1}(\boldsymbol{x})\} \tag{2}$$

Suppose $F_{k,c}$ and $F_{k,I}$ are the sets of the foreground pixels identified using chromaticity- and intensity-based background models, respectively. The set of final foreground pixels can be computed as the intersection between chromaticity-based foreground (dilated) and intensity-based foreground:

$$F_k = (F_{k,c} \oplus B) \cap F_{k,I} \tag{3}$$

where $\oplus$ denotes the morphological dilation and B is the structuring element for dilation. The fusion between two types of mixture models overcomes the disadvantages of using each model separately[8]. Intensity-based detection is sensitive to illumination changes; while a foreground region detected using only chromaticity may be split when a part of the underlying target has a chromaticity similar to the background. In addition, chromaticity-

based detection is sensitive to noise in poorly-lit regions. By using both models simultaneously, in regions with lighting variation, few spurious chromaticity-based foreground regions are produced, which masks spurious intensity-based foregrounds. In poorly-lit regions, few spurious intensity-based foreground regions are detected, which masks the spurious chromaticity-based foregrounds. Split or shrinking chromaticity-based foregrounds, due to their similar chromaticity to background, can be bridged or re-sized by the morphological dilation of B. In comparison with intensity-based models, this detection scheme has been shown to greatly reduce the false positive rate in PETS'2001 Dataset 3, which has significant illumination variation[3].

The foreground pixels are filtered by a morphological closing (dilation-plus-erosion) operation and then clustered into foreground regions using a connected component analysis. A minimum number of foreground pixels is set for each region to rule out small disturbances. A foreground region may correspond to an object, a group of objects due to dynamic occlusion, or part of an object due to static occlusion. It is represented by a foreground measurement vector, $\boldsymbol{f}=[r_c \quad c_c \quad r_1 \quad c_1 \quad r_2 \quad c_2]^{\mathrm{T}}$, where (r_c, c_c) is the centroid, (r_1, c_1) and (r_2, c_2) are the two opposite corners of the bounding box. r_1, c_1, r_2, c_2 represent the top, left, bottom and right bounding edges, respectively($r_1<r_2, c_1<c_2$). In this paper, we use $\boldsymbol{f}(i)$ to represent the i-th element of the vector $\boldsymbol{f}$, e. g. $\boldsymbol{f}(1)=r_c$. The bounding box of each foreground blob covers a rectangular region defined by $R_f=\{(r,c): r\in[r_1, r_2], c\in[c_1, c_2]\}$.

3 Object dynamics mode

A Kalman filter[9] based on a first-order motion model is used to track each object according to the object measurement vector, $\boldsymbol{z}=[r_c \quad c_c \quad r_1 \quad c_1 \quad r_2 \quad c_2]^{\mathrm{T}}$. We distinguish object measurements from foreground measurements, because they are the same only for separate objects. Because our system aims to monitor pedestrians and vehicles, each target is assumed to move along a linear trajectory at constant velocity and with constant size. In practice, any minor violation of this assumption can be encoded in the process covariance matrix. The state vector used is $\boldsymbol{x}=[r_c \quad c_c \quad \dot{r}_c \quad \dot{c}_c \quad \Delta r_1 \quad \Delta c_1 \quad \Delta r_2 \quad \Delta c_2]^{\mathrm{T}}$, where $(\Delta r_1, \Delta c_1)$ and $(\Delta r_2, \Delta c_2)$ are the relative positions of the two opposite bounding box corners to the centroid. They not only incorporate height and width information, but also accurately represent the bounding box even when the centroid is shifted away from the geometric centre of the bounding box, e. g. due to asymmetry or shadows.

The state transition and measurement equations are:

$$\begin{aligned} \boldsymbol{x}_k &= A\boldsymbol{x}_{k-1}+\boldsymbol{w}_{k-1} \\ \boldsymbol{z}_k &= H\boldsymbol{x}_k+\boldsymbol{v}_k \end{aligned} \tag{4}$$

$\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are process noise and measurement noise, respectively, and $\boldsymbol{w}_k\sim N(\boldsymbol{0}, Q_k)$, $\boldsymbol{v}_k\sim N(\boldsymbol{0}, R_k)$; the state transition matrix A and measurement matrix H are:

$$A=\begin{bmatrix} I_2 & \Delta T I_2 & O_2 & O_2 \\ O_2 & I_2 & O_2 & O_2 \\ O_2 & O_2 & I_2 & O_2 \\ O_2 & O_2 & O_2 & I_2 \end{bmatrix}, \quad H=\begin{bmatrix} I_2 & O_2 & O_2 & O_2 \\ I_2 & O_2 & I_2 & O_2 \\ I_2 & O_2 & O_2 & I_2 \end{bmatrix} \tag{5}$$

where I_2 and O_2 are 2×2 identity and zero matrices; ΔT is the time interval between frames. The a priori estimate $\hat{\boldsymbol{x}}_k^-$ and a posteriori estimate $\hat{\boldsymbol{x}}_k^+$ are iteratively computed by

$$\begin{aligned} \hat{\boldsymbol{x}}_k^- &= A\,\hat{\boldsymbol{x}}_{k-1}^+ \\ \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + K_k(\boldsymbol{z}_k - H\,\hat{\boldsymbol{x}}_k^-) \end{aligned} \tag{6}$$

where K_k is the Kalman gain matrix that is sought to minimize the a posteriori error covariance P_k^+ in a least-square sense and can also be computed with P_k^+ and the a priori error covariance P_k^- in an iterative way:

$$\begin{aligned} P_k^- &= AP_{k-1}^+ A^{\mathrm{T}} + Q_{k-1} \\ K_k &= P_k^- H^{\mathrm{T}} [HP_k^- H^{\mathrm{T}} + R_k]^{-1} \\ P_k^+ &= [I - K_k H] P_k^- \end{aligned} \tag{7}$$

4 Scene model

Because the camera is fixed, a scene model can be constructed for a specific camera position. Whilst this is currently done manually, some previous work on path learning[10] may be extended to derive an automatic method for learning the scene model. This model helps reasoning about the termination and occlusion of objects by scene elements. Three types of static occlusions in a scene are identified (Fig. 1).

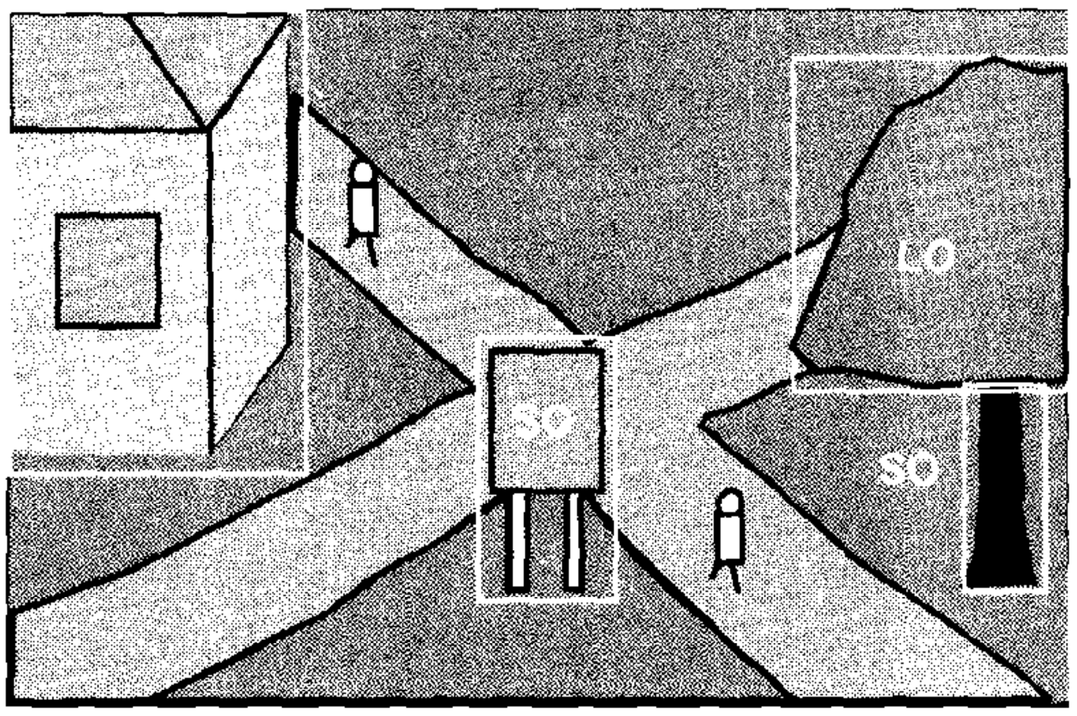


Fig. 1 Long-term occlusions (*LO*) and short-term occlusions (SO) in a scene

- **Border occlusions (BO)**, outside the limits of the camera field-of-view (FOV).
- **Long-term occlusions (LO)**, where objects may leave the scene earlier than expected, corresponding to the termination of a record in the object database. The long-term occlusion may exist at the border (e. g. buildings or vegetation) or in the middle of an image (e. g. at the doors of a building).
- **Short-term occlusions (SO)**, where an object may be temporarily occluded by a static occlusion, e. g. a tree or a road sign. Prior knowledge of these occlusions helps avoid missing existing objects and creating "new" objects.

Each occlusion is characterized by its type (BO, LO or SO) and the region (R_{BO}, R_{LO}, or R_{SO}) defined by its bounding box. The definitions of R_{LO} and R_{SO} are similar to that of R_f, including the internal region of the underlying bounding box, whilst R_{BO} covers the region outside the field-of-view:

$$R_{BO} = \{(r,c): r \notin [r_{\min}, r_{\max}], \quad c \notin [c_{\min}, c_{\max}]\}$$

The overlap of these static occlusions with the predicted centroid of an object can be used to predict object terminating (exiting) and occlusion. After the a priori estimate of the state is determined, each object is subject to the status prediction based on the scene model and predictive measurement:

$$\hat{\boldsymbol{z}}_k^- = H\hat{\boldsymbol{x}}_k^- \tag{8}$$

An object is labelled as PREDICT_TERMINATED, if its predicted centroid is within the border occlusion (BO) or a long-term occlusion (LO), i. e. $(\hat{\boldsymbol{z}}_k^-(1), \hat{\boldsymbol{z}}_k^-(2)) \in R_{BO} \bigcup (\bigcup_{n=1}^{N_{LO}} R_{LO,n})$. An object is labelled as PREDICT_OCCLUDED, if its predicted centroid is within a short-term occlusion (SO), i. e. $(\hat{\boldsymbol{z}}_k^-(1), \hat{\boldsymbol{z}}_k^-(2)) \in \bigcup_{n=1}^{N_{SO}} R_{SO,n}$. Currently a rectangular bounding box is used for each static occlusion to minimize the computational cost.

5 Partial observability

For tracking multiple objects in a complex scene, it is noted that the object measure-

ment $\mathbf{z}_k$ may be either partly or completely unavailable. This occurs due to dynamic occlusion between objects, static occlusion, or just the failure of foreground detection. Fig. 2 shows some examples of partial observation.

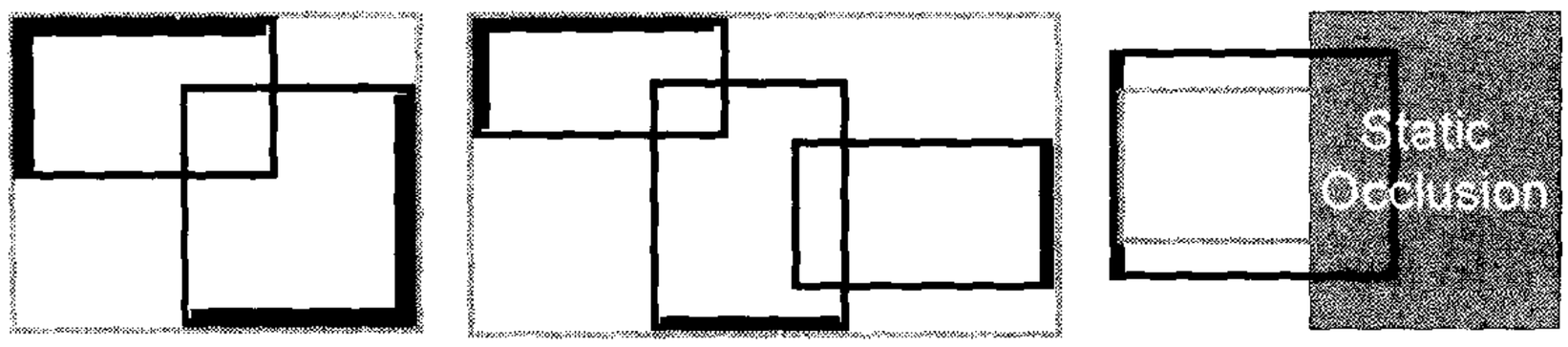


Fig. 2 Partial observations when objects are grouped (a)(b) or behind a static occlusion (c). Grey lines represent the foreground bounding boxes; thick and thin black lines represent observable and unobservable bounding edges of objects, respectively.

5.1 Deciding observability

We decide the observability of the objects based on the predictive measurement $\hat{\mathbf{z}}_k^-$, the foreground measurement $\boldsymbol{f}_k$ and the scene model. The outcome is represented by the observability vector $\boldsymbol{m}_k$, which has the same dimensions as the object measurement vector $\boldsymbol{z}_k$, with a one-to-one correspondence in their elements. Each element of $\boldsymbol{m}_k$ has only two possible values: 1 for OBSERVABLE and 0 for UNOBSERVABLE. At the beginning of iteration (time) k, each object is reset to OBSERVABLE, i. e. $\boldsymbol{m}_k(l) = OBSERVABLE$, $l \in [1,6]$, and then subject to a 3-stage modification.

1) Observability in grouping (Figs. 2(a) and (b))

For the i-th tracked object ($i \in [1, N_o]$) and j-th foreground measurement ($j \in [1, N_f]$), a match score based on Mahalanobis distance is computed as:

$$D(i,j) = \begin{cases} 0, & \text{if } (\hat{\boldsymbol{z}}_{k,i}^-(1),\ \hat{\boldsymbol{z}}_{k,i}^-(2)) \in R_{f,j} \\ (\boldsymbol{f}_{k,j} - \hat{\boldsymbol{z}}_{k,i}^-)^{\mathrm{T}} (HP_{k,i}^- H^{\mathrm{T}} + R_k)^{-1} (\boldsymbol{f}_{k,j} - \hat{\boldsymbol{z}}_{k,i}^-), & \text{otherwise} \end{cases} \tag{9}$$

Each object selects its corresponding foreground measurement in terms of minimum match score within a tolerance ε. For the j-th foreground measurement, the objects that best correspond to it form a group of objects that are most likely to be merged:

$$G_j = \{i \in [1, N_o] : j = \arg\min_{n \in [1, N_f]} D(i,n),\ D(i,j) < \varepsilon\} \tag{10}$$

G_j may include multiple objects (multiple-to-one correspondence), one object (one-to-one correspondence), or be empty (no correspondence). For each object $p \in G_j$, its observability is modified according to whether its bounding edge(s) is (are) also the bounding edge(s) of the group:

$$\begin{aligned} \boldsymbol{m}_{k,p}(3) &= UNOBSERVABLE \quad \text{if } \hat{\boldsymbol{z}}_{k,p}^-(3) > \min_{q \in G_j} \{\hat{\boldsymbol{z}}_{k,q}^-(3)\} \\ \boldsymbol{m}_{k,p}(4) &= UNOBSERVABLE \quad \text{if } \hat{\boldsymbol{z}}_{k,p}^-(4) > \min_{q \in G_j} \{\hat{\boldsymbol{z}}_{k,q}^-(4)\} \\ \boldsymbol{m}_{k,p}(5) &= UNOBSERVABLE \quad \text{if } \hat{\boldsymbol{z}}_{k,p}^-(5) < \max_{q \in G_j} \{\hat{\boldsymbol{z}}_{k,q}^-(5)\} \\ \boldsymbol{m}_{k,p}(6) &= UNOBSERVABLE \quad \text{if } \hat{\boldsymbol{z}}_{k,p}^-(6) < \max_{q \in G_j} \{\hat{\boldsymbol{z}}_{k,q}^-(6)\} \end{aligned} \tag{11}$$

2) Observability of foreground measurement

The observability of object $p \in G_j$ also depends on the observability of its associated foreground measurement j. If the bounding box of the underlying foreground region touches the border of the field-of-view, its relevant bounding edge becomes UNOBSERVABLE and thus inhibits (masks) the relevant observability for the associated object bounding edge, i. e. :

$$\begin{aligned} \boldsymbol{m}_{k,p}(3) &= UNOBSERVABLE \quad \text{if } \boldsymbol{f}_{k,j}(3) = r_{\min} \\ \boldsymbol{m}_{k,p}(4) &= UNOBSERVABLE \quad \text{if } \boldsymbol{f}_{k,j}(4) = c_{\min} \\ \boldsymbol{m}_{k,p}(5) &= UNOBSERVABLE \quad \text{if } \boldsymbol{f}_{k,j}(5) = r_{\max} \\ \boldsymbol{m}_{k,p}(6) &= UNOBSERVABLE \quad \text{if } \boldsymbol{f}_{k,j}(6) = c_{\max} \end{aligned} \tag{12}$$

3) Observability in occlusion (Fig. 2(c))

When an object is partly hidden behind a static occlusion, the measurement of some of its bounding edges becomes unreliable. For the i-th object, each of its predicted bounding edges is checked. If either corner delimiting that edge is within a static occlusion defined in the scene model, that edge becomes UNOBSERVABLE, i. e.:

$$\begin{aligned}
&\boldsymbol{m}_{k,i}(3) = \boldsymbol{m}_{k,i}(4) = UNOBSERVABLE \quad \text{if } (\hat{\boldsymbol{z}}_{k,i}^{-}(3), \hat{\boldsymbol{z}}_{k,i}^{-}(4)) \in \Omega \\
&\boldsymbol{m}_{k,i}(3) = \boldsymbol{m}_{k,i}(6) = UNOBSERVABLE \quad \text{if } (\hat{\boldsymbol{z}}_{k,i}^{-}(3), \hat{\boldsymbol{z}}_{k,i}^{-}(6)) \in \Omega \\
&\boldsymbol{m}_{k,i}(5) = \boldsymbol{m}_{k,i}(4) = UNOBSERVABLE \quad \text{if } (\hat{\boldsymbol{z}}_{k,i}^{-}(5), \hat{\boldsymbol{z}}_{k,i}^{-}(4)) \in \Omega \\
&\boldsymbol{m}_{k,i}(5) = \boldsymbol{m}_{k,i}(6) = UNOBSERVABLE \quad \text{if } (\hat{\boldsymbol{z}}_{k,i}^{-}(5), \hat{\boldsymbol{z}}_{k,i}^{-}(6)) \in \Omega
\end{aligned} \tag{13}$$

where $\Omega = R_{BO} \bigcup (\bigcup_{n=1}^{N_{LO}} R_{LO,n}) \bigcup (\bigcup_{n=1}^{N_{SO}} R_{SO,n})$.

After the observability of the four bounding edges for i-th object is determined, the observability of its centroid can be decided as OBSERVABLE only when all the four bounding edges are OBSERVABLE, i. e.:

$$\boldsymbol{m}_{k,i}(1) = \boldsymbol{m}_{k,i}(2) = \prod_{l=3}^{6} \boldsymbol{m}_{k,i}(l) \tag{14}$$

5.2 Using observability

For a partially unobservable object, a measurement vector is constituted, whose members can be classified into two inter-correlated blocks (r_c, r_1, r_2) and (c_c, c_1, c_2). The inter-block variables are bound by the constant height (Δr_1 and Δr_2) and constant width (Δc_1 and Δc_2) assumption. Within each block, if all the variables are unobservable, the only clue for their measurements are the prediction; if part of its variables are observable, the unobservable measurements can be jointly deduced from the observable measurements and prediction. Suppose the observability matrix M_k is a diagonal matrix whose main diagonal is the observability vector $\boldsymbol{m}_k$, i. e.:

$$M_k(i,j) = \begin{cases} \boldsymbol{m}_k(i), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

The measurement vector is estimated by:

$$\boldsymbol{z}_k = M_k \boldsymbol{f}_k + (I - M_k)[\alpha \boldsymbol{d}_k + (1-\alpha)\hat{\boldsymbol{z}}_k^{-}] \tag{16}$$

where $\boldsymbol{d}_k$ is the directly deduced measurements of unobservable variables from observable measurements using constant height and width assumption, and α controls the combination weights between the directly deduced measurements and the prediction ($0 \leqslant \alpha \leqslant 1$). If all the variables in an inter-correlated block are unobservable, $\boldsymbol{d}_k = \hat{\boldsymbol{z}}_k^{-}$ for that block and this is equivalent to $\alpha = 0$. The height and width information in the a priori state estimate $\hat{\boldsymbol{x}}_k^{-}$ is used to compute $\boldsymbol{d}_k$.

Because the "measurement" for an unobservable variable is estimated rather than an actual measurement, the corresponding element in the measurement covariance matrix R_k has to be increased by λ times ($\lambda > 1$) to reflect increased uncertainty. Suppose R is the measurement covariance matrix of a completely observable object, the measurement covariance matrix for a partially observable object becomes:

$$R_k = M_k R M_k^{\mathrm{T}} + (I - M_k)\lambda R (I - M_k)^{\mathrm{T}} \tag{17}$$

Because the centroid and bounding edges of each object are measured independently, R is a diagonal matrix. In addition, M_k is diagonal as well, with each element being either 1 or 0. Therefore, the equation above can be simplified as:

$$R_k = [M_k + \lambda(I - M_k)]R \tag{18}$$

Using this equation, an observable variable of an object is updated with a normal covariance value, whilst an unobservable variable is updated with a larger covariance value. Therefore, the observable bounding edges contribute more to the object estimation than the unobservable bounding edges.

In the correspondence of the existing objects and measured foreground regions (Eq. (10)), it is possible that some objects cannot find a match with any foreground region. If labelled with PREDICT_TERMINATED, they are assumed to be terminated; if labelled with PREDICT_OCCLUDED, they are assumed to be behind some static occlusion and updated using the predictive state, $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^-$; otherwise, they are assumed to be missing due to foreground detection failure, updated using the predictive state, and subject to termination if still unmatched in the following N frames. It is also possible that no existing object corresponds to a detected foreground region. In this case, a new object is created and its state is initialised by $\boldsymbol{f}_k$ and a zero velocity.

6 Results

To evaluate the performance of our tracking algorithm, we have tested it on a range of image sequences and compared it with other two algorithms using blind tracking through occlusion. To distinguish the effect of using partial observation, both algorithms were designed to be the same as the new one (e. g. the same Kalman tracker, same state and measurement vectors), except their treatment to objects in grouping or occlusion:

• Algorithm 1—The object with the smallest Mahalanobis distance to the group foreground measurement is estimated with the measurement of the group, $\boldsymbol{z}_k = \boldsymbol{f}_k$; the others are updated using prediction, i. e. $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^-$, as in [3].

• Algorithm 2—All the objects in a group are updated using prediction, i. e. $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^-$, as in [2].

The image sequences used for the demonstration in this paper are the testing Dataset[1] 1 (CAM1 and CAM2) for PETS'2001. We processed frames 1 to 2681 at a temporally sub-sampled rate of 5 (simulating 5 fps) and at the half frame size (384×288). $N = 5$ in our experiments, which is determined to balance between missing true objects for small N and maintaining many phantom objects for large N. In the image results shown below, black and white boxes represent foreground measurement $\boldsymbol{f}_k$ and object a posteriori estimate $\hat{\boldsymbol{x}}_k^+$, respectively. A white dotted box represents a partly or completely unobservable object. A white curve represents the centroid trajectory of an active object.

6.1 Qualitative performance

Fig. 3 shows an example of tracking through occlusion. In this example, a group of people (object 4) walk toward and then pass by a stationary car (object 2). At frames 946-991, both the objects are grouped and segmented as a large foreground blob (Fig. 3(b)). Algorithm 1 matches the group foreground blob with object 4, the size of which is then gradually adapted to that of the group measurement. After the group of people is split from the car and segmented as a separate, smaller foreground blob (Fig. 3(c)), Algorithm 1 rejects the match between this blob and object 4, due to the great difference in size and position, and creates a new object (Object 6). Therefore, the group of people changes its label after the occlusion. By using the partial observation, the location and size of object 4 is more accurately estimated during occlusion (Fig. 3(e)). Finally, object 4 is correctly matched to the separate foreground blob (Fig. 3(f)).

Fig. 4 shows another example of tracking through occlusion. In this example, a dark car (object 11 in top row and object 10 in bottom row) moves toward a stationary white van (object 3) and finally occludes it. At frames 2246-2496, both the targets are segmented as a large foreground blob. Algorithm 2 uses linear prediction to update the estimate of object 11 during the grouping. Because object 11 moves in a non-linear trajectory, there exist some errors between the estimate and the foreground blob measurement (Fig. 4(b),

1) PETS'2001, http://www.visualsurveillance.org/PETS2001

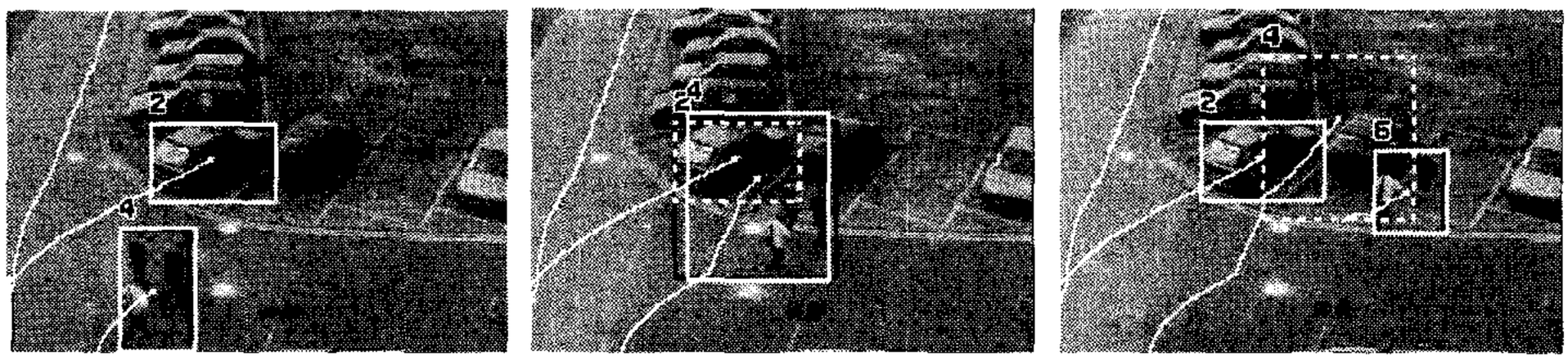


Fig. 3 Example 1 of object tracking through occlusion, (a)～(c) using Algorithm 1, and (d)～(f) using partial observation($\alpha=0$)

see the unfitted bottom and right bounding edges). These estimation errors accumulate and object 11 finally fails to match the corresponding foreground blob (Fig. 4(c)). Using the partial observation, the bottom and right bounding edges of object 10 closely fit to the foreground blob (Figs. 4(e) and (f)). Object 10 even has a non-linear trajectory during grouping (Fig. 4(f)), which indicates the linear motion model has been continuously adapted to the non-linear motion. It is also noted that the estimate of the top bounding edge of object 10 is not accurate (Fig. 4(f)), because it has been unobservable for a long time.

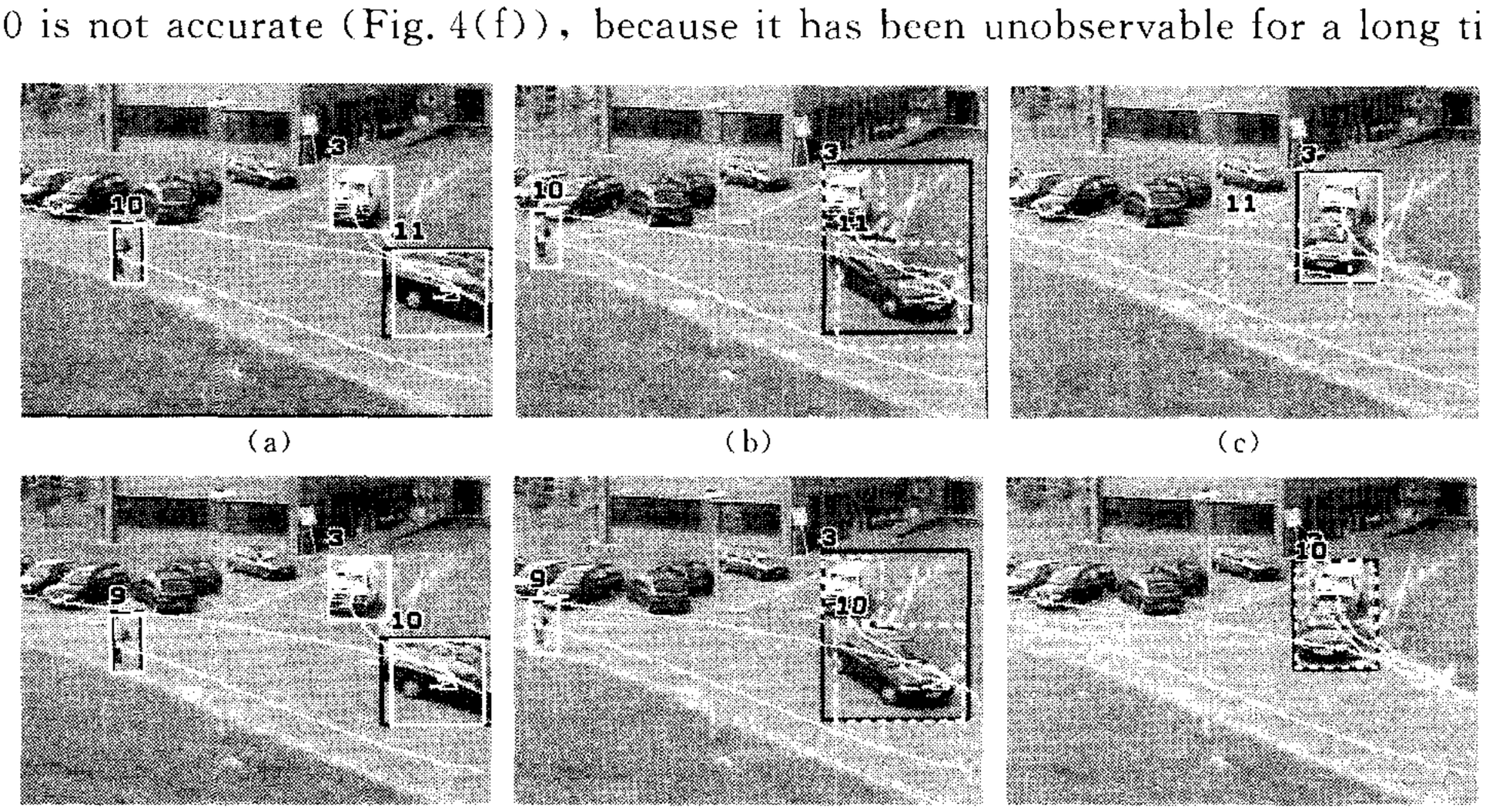


Fig. 4 Example 2 of object tracking through occlusion, (a)～(c) using Algorithm 2 and (d)～(f) using partial observation($\alpha=0$)

For all the 9 grouping events (19 objects involved) in which objects merge and then split, we counted the mis-tracking events in which any object in a group changes its label after splitting. The result shown in Table 1 indicates that the new algorithm performs more reliably than Algorithms 1 and 2.

Table 1 Counts of erroneous tracking in 9 grouping-and-splitting events

	Algorithm 1	Algorithm 2	New ($\alpha=0$)	New ($\alpha=1$)
Count of Errors	1	2	0	0

6.2 Quantitative performance

The advantages of partial observation not only are reflected in the qualitative comparisons as above, but also exist in some quantitative measures applied to the tracking results in which both Algorithms 1 and 2 succeed. The first measure is the tracking error between actual and predictive measurements, i. e.

$$e_k = \| z_k - \hat{z}_k^- \| \tag{19}$$

For objects updated using prediction, this error is set to zero.

The second quantitative measure is the path coherence, which represents a measure of agreement between the derived object trajectory and the motion smoothness constraints[11]. Suppose s_k is the segment vector between the centroid estimates at two consecutive frames,

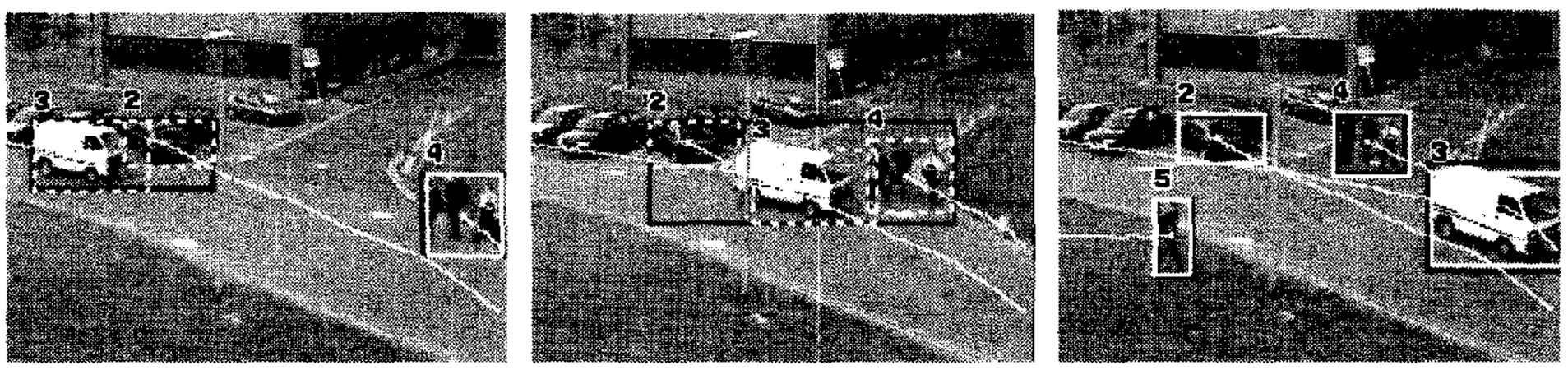
$$s_k = [\hat{x}_k^+(1) - \hat{x}_{k-1}^+(1) \quad \hat{x}_k^+(2) - \hat{x}_{k-1}^+(2)] \tag{20}$$

The path coherence function used is:

$$\Phi_k = w_1 \left[1 - \frac{|s_k \cdot s_{k+1}|}{\|s_k\| \|s_{k+1}\|}\right] + w_2 \left[1 - 2\frac{\sqrt{\|s_k\| \|s_{k+1}\|}}{\|s_k\| + \|s_{k+1}\|}\right] \tag{21}$$

where the weights w_1 and w_2 control the importance of direction coherence and velocity coherence ($w_1 = 0.5$ and $w_2 = 0.5$ in this paper), and $\Phi_k \in [0, 1]$. A successful tracking scheme usually generates low values in tracking error and path coherence function.

These two quantitative measures were selected because they are the basis of most existing motion correspondence algorithms that usually assume the smoothness of motion. These measures are demonstrated using the example shown in Fig. 5, which is overlaid by the tracking result using partial observation. In this example, a white van (object 3) first passes by a newly stationary dark car (object 2), heads toward and occludes a group of people (object 4), decelerates and stops separately at the right border of the FOV. Object 3 has been grouped at frames 806-941. Due to the linear trajectories for the objects involved, Algorithms 1 and 2 also succeed in this example. However, these two algorithms have different performance based on our quantitative measures.



(a) (b) (c)

Fig. 5 Example 3 of object tracking using partial observation($\alpha = 0$)

Fig. 6 shows the tracking errors and path coherence values for object 3 in Fig. 5, resulting from all the three algorithms. There are two points that should be noted. Firstly, the centroid estimation error only accounts for about one third of the entire tracking error, because the latter also includes the errors for two bounding corners. Secondly, the zero values in the tracking error and coherence function for Algorithms 1 and 2 arise from grouping and state updating using prediction, representing uncertainty rather than perfect tracking. Therefore to be fair, our comparison is concentrated on the measures just after the end of grouping (frame 946). At that time objects 3 and 4 split and are re-tracked; the tracking error and coherence are expected to have a peak value.

For the 12 objects involved in all the 6 grouping-and-splitting events in which all the three algorithms succeed, the peak values of the new algorithm are lower than those of Algorithms 1 and 2 in each case; the average peak values are shown in Table 2. The quantitative measures for the new algorithm are much lower than those for Algorithms 1 and 2,

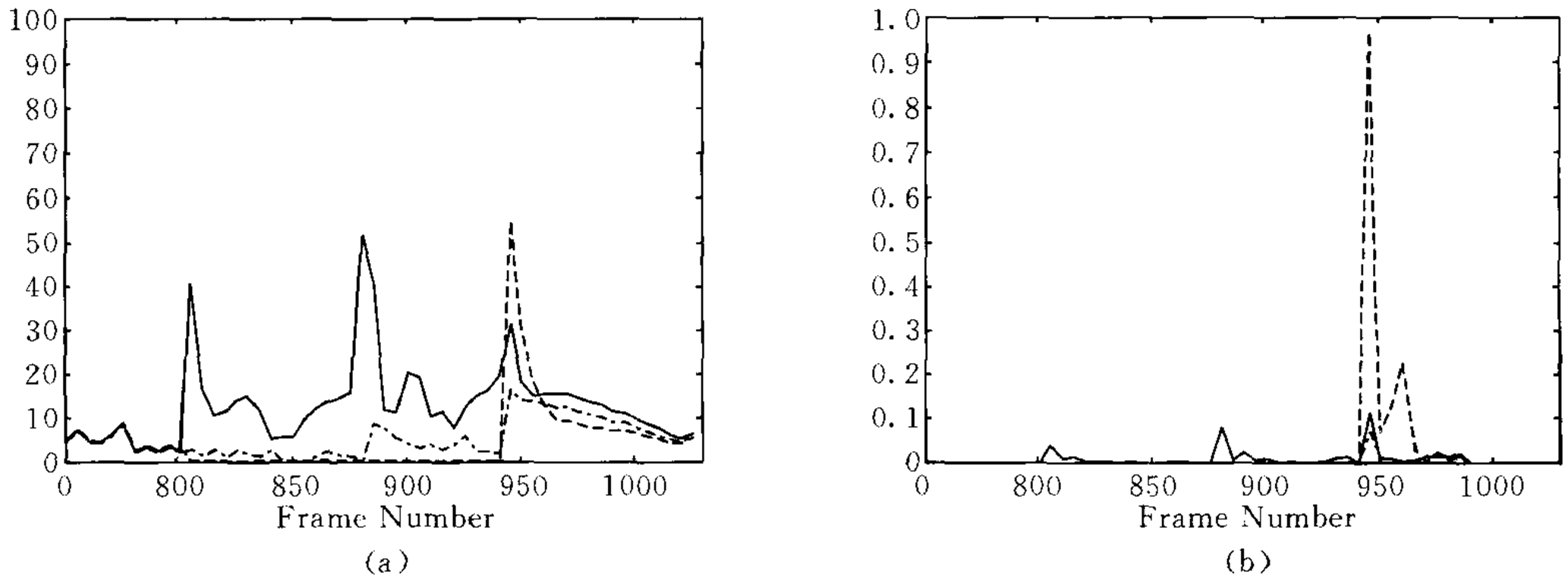


Fig. 6 (a) Tracking errors in pixels and (b) path coherence, for object 3 in Fig. 5, using Algorithm 1 (solid lines), Algorithm 2 (dash lines), and partial observation ($\alpha=0$, dot-dash lines)

indicating its improved performance. The reason is that even fed by partial observation only during grouping, objects could deduce their unobservable bounding edges according to the built-in relation among members of the measurement vector. For example, on the assumption of constant size, the left and right edges of an object should share a horizontal velocity, and the top and bottom edges should share a vertical velocity. The deduction of unobservable variables can be either direct ($\alpha=1$) or implicit when the Kalman filter seeks the optimal solution for the a posteriori estimate ($\alpha=0$). In both the cases, the measurements of the observable variables are propagated to the unobservable variables. Therefore, even with an incomplete measurement input, the objects still have the estimates of all the four bounding edges adapted to the new, partial measurement. This is partly reflected by the non-zero tracking errors of object 3 during grouping (Fig. 6(a)), which prevents the tracking errors from accumulating and makes object 3 adaptive to the deceleration. The after-grouping peak measures of the new algorithms using $\alpha=1$ fluctuate around those using $\alpha=0$. Their relative values depend on whether the constant size assumption ($\alpha=1$) or the constant velocity and size assumption ($\alpha=0$) is a better fit to the practical situations in the testing sequences.

Table 2 Quantitative measures of the tracking algorithms

	Algorithm 1	Algorithm 2	New ($\alpha=0$)	New ($\alpha=1$)
Tracking errors	28.27	29.96	19.33	16.81
Path coherence	0.2765	0.2461	0.0923	0.0863

7 Conclusions

We have presented a tracking algorithm utilizing partial observation of each target through grouping or occlusion. The unobservable variables can be estimated by a Kalman filter based on the measurement of observable variables, the state prediction, as well as the scene model. This makes target estimation adaptive to small changes of direction and accelerations during grouping or occlusion. The new algorithm has advantages over traditional blind tracking schemes in terms of 32% decrease in tracking error and 63% decrease in path coherence value, which leads to smoother object trajectories. It is noted that the improvement is still under-estimated because we only accounted for the cases in which both Algorithms 1 and 2 maintain the tracking successfully. These benefits have also been demonstrated in the on-line 2D tracking systems as part of the IMCASM (Intelligent Multi-Camera Surveillance and Monitoring) project.

Acknowledgements The authors would like to thank James Black for the discussion on the Kalman filter.

References

1 Intille S S, Davis J W, Bobick A F. Real-time closed-world tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Juan: IEEE Computer Society. 1997. 697～703

2 Rosales R, Sclaroff S. Improved tracking of multiple humans with trajectory prediction and occlusion modelling. In: Proceedings of IEEE Workshop on the Interpretation of Visual Motion, Santa Barbara: IEEE Computer Society, 1998. 117～123

3 Ellis T, Xu M. Object detection and tracking in an open and dynamic world. In: Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance, Kauai: IEEE Computer Society, 2001. 31～38

4 Xu M, Ellis T. Partial observation vs. blind tracking through occlusion. In: Proceedings of British Machine Vision Conference, Cardiff: BMVA, 2002. 777～786

5 Mammen J P, Chaudhuri S, Agrawal T. Simultaneous tracking of both hands by estimation of erroneous observations. In: Proceedings of British Machine Vision Conference, Manchester: BMVA, 2001. 83～92

6 Dockstader S L, Tekalp A M. Tracking multiple objects in the presence of articulated and occluded motion. In: Proceedings of IEEE Workshop on Human Motion, Austin: IEEE Computer Society, 2000. 88～95

7 Stauffer C, Grimson W E. Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins: IEEE Computer Society, 1999. 246～252

8 Xu M, Ellis T. Illumination-invariant motion detection using colour mixture models. In: Proceedings of British Machine Vision Conference, Manchester: BMVA, 2001. 163～172

9 Kalman R E. A new approach to linear filtering and prediction problems. *Transcation of ASME—Journal of Basic Engeering*, 1960, **82-D**: 35～45

10 Makris D, Ellis T. Path detection in video surveillance. *Image and Vision Computing*, 2002, **20**(12): 895～903

11 Sethi I K, Jain R. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on PAMI*, 1987, **9**(1): 56～73

Ming Xu Received his bachelor degree(1989) and master degree(1992) from Xi'an Jiaotong University of P. R. China in communication engineering and received the Ph. D. degree (2001) in computer vision from University of Birmingham, UK. He worked with Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University (1992～1996) and Information Engineering Centre at City University, London (1999～2002), before joining Digital Imaging Research Centre at Kingston University, UK. Dr. Xu is a member of British Machine Vision Association (BMVA) and was awarded a Science and Technology Progress Prize by the Ministry of Education, P. R. China in 1997. His research interests include motion analysis and tracking, multi-view techniques, scale-space analysis.

Tim Ellis Received his bachelor degree in physics from the University of Kent at Canterbury in 1974 and his Ph. D. degree in biophysics from London University in 1981. He joined City University in 1979 as a research fellow, investigating algorithms for surface inspection. In 1984 he was awarded a five year Advanced Fellowship by the SERC in the field of intelligent instrumentation. In 1989 he was appointed lecturer in the Department of Electrical, Electronic and Information Engineering at City University, and is currently a Reader and Director of the Information Engineering Centre. He is leader of the Machine Vision Group, and past Chairman of the British Machine Vision Association. His research interests include development of algorithms for extracting structural primitives from images, analysis of motion in image sequences, colour image processing and hardware for image processing. The research is applied to problems of video surveillance and monitoring, object tracking, pose determination and automatic inspection.

利用部分观测跟踪被遮挡的目标

Ming Xu Tim Ellis

(*Information Engineering Centre, City University, London, EC1V 0HB*, 英国)

(E-mail: {m. xu, t. j. ellis}@city. ac. uk)

摘 要 提出了一个在单个固定摄像机下进行多目标跟踪的方法. 利用亮度和色度混合模型和卡尔曼滤波器来检测跟踪目标,为了利于预测和解释被遮挡的物体,建立了场景的模型. 在遮挡的情况下,和传统的盲跟踪不同,本文中的目标状态是由可用的部分观测来估计的. 对目标的观测取决于预测、前景观测和场景模型,这使得本文算法在定性或定量的分析下都表现出更加鲁棒的性能.

关键词 部分观测,场景模型,前景区域

中图分类号 TP391.41