ELSEVIER

Image and Vision Computing 21 (2003) 913–929

image AND vision COMPUTING

www.elsevier.com/locate/imavis

A real time adaptive visual surveillance system for tracking low-resolution colour targets in dynamically changing scenes

Pakorn KaewTrakulPong[a,*], Richard Bowden[b]

[a]INC, King Mongkut's University of Technology Thonburi, 91 Prachauthit Road (Suksawad 48) Bangmod, Thung Kharu District, Bangkok 10140, Thailand
[b]CVSSP, ECM, University of Surrey Guildford, Surrey GU2 7XH, UK

Received in revised form 29 November 2002; accepted 20 March 2003

Abstract

This paper presents a variety of probabilistic models for tracking small-area targets which are common objects of interest in outdoor visual surveillance scenes. We address the problem of using appearance and motion models in classifying and tracking objects when detailed information of the object's appearance is not available. The approach relies upon motion, shape cues and colour information to help in associating objects temporally within a video stream. Unlike previous applications of colour and complex shape in object tracking, where relatively large-size targets are tracked, our method is designed to track small colour targets commonly found in outdoor visual surveillance. Our approach uses a robust background model based around online Expectation Maximisation to segment moving objects with very low false detection rates. The system also incorporates a shadow detection algorithm which helps alleviate standard environmental problems associated with such approaches. A colour transformation derived from anthropological studies to model colour distributions of low-resolution targets is used along with a probabilistic method of combining colour and motion information. A data association algorithm is applied to maintain tracking of multiple objects under circumstances. Simple shape information is employed to detect subtle interactions such as occlusion and camouflage. A novel guided search algorithm is then introduced to facilitate tracking of multiple objects during these events. This provides a robust visual tracking system which is capable of performing accurately and consistently within a real world visual surveillance arena. This paper shows the system successfully tracking multiple people moving independently and the ability of the approach to maintain trajectories in the presence of occlusions and background clutter.

Keywords: Visual surveillance; Multiple target tracking; Tracking through occlusion; Low resolution target; Stochastic search; Perceptualised colour; Particle filter

1. Introduction

There has been extensive work on the subject of tracking multiple point targets, typically in the arena of radar tracking. The main components of which are the tracking process itself and data association. Tracking deals with maintaining motion models of the objects being tracked whereas data association uses the motion model which summaries all past measurements of a target to predict a position for the next time step. Data association is then responsible for matching or assigning measurements at the current time to targets. As a number of objects move independently, target observations may fall in other targets' predicted areas. False or undetected measurements further introduce ambiguity to this assignment problem. Unlike Radar tracking systems, visual-based systems require preprocessing to obtain measurements of a target motion state. Targets may be occluded by some stationary objects in the scene as well as by other targets. Occlusions do occur in Radar tracking systems, however, they are few and do not last long. Therefore, standard tracking and data association algorithms are sufficient. In visual surveillance, targets consisting of pedestrians and vehicles can have relatively slow non-linear motions. The occlusions tend to happen more frequently and last for longer periods. Standard tracking and data association algorithms may terminate their tracks sooner to reduce the chance of incorrect assignment (as the prediction uncertainty grows with time). In appearance-based tracking systems, a reliable

* Corresponding author.
E-mail addresses: ipakpong@kmutt.ac.th (P. KaewTrakulPong), pakorn.kaewtrakulpong@brunel.ac.uk (P. KaewTrakulPong), r.bowden@surrey.ac.uk (R. Bowden).

0262-8856/03/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S0262-8856(03)00076-3

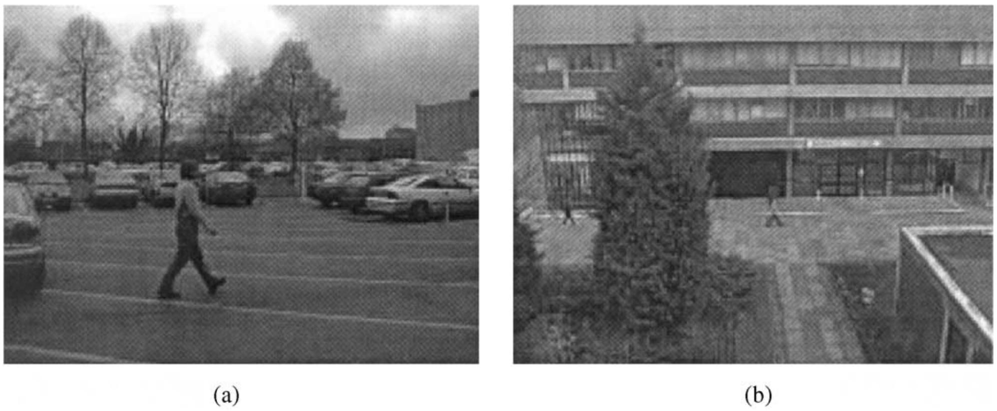

Fig. 1. (a) Scene with a high-resolution target, (b) scene with a low-resolution target.

model is required to facilitate tracking an object through background clutter that overcomes these aforementioned difficulties.

This paper addresses the problem of using appearance and motion models in tracking low-resolution colour objects. Instead of relying solely upon a motion model and maintaining multiple hypotheses, simple shape and colour information can be useful in data association resulting in reducing the number of hypotheses that need be supported. However, the number of pixels constituting an object can be too small to be able to build a reliable appearance model for either shape or colour tracking. An example of targets used in systems by many authors is shown in Fig. 1(a) while our method monitors targets in the scene similar to Fig. 1(b). In such cases, the colour distribution learnt from the scene is deemed unreliable due to the limited supporting evidence obtained from the scene. The number of pixels supporting the object is too few to train a complex shape or colour model. As the model becomes more complex, the number of required training samples increases exponentially. This not only leads to overfitting, but some algorithms may converge upon singular solutions leading to an unstable system. Under these circumstances, most systems treat the colour information as unreliable and use motion cues and simple shape features alone to track objects [10,18,25]. This paper utilises both simple shape and motion along with a colour model based on transformations derived from psychophysical studies [24,26]. This transformation provides the ability to construct a simple colour profile based upon a small sample data set which overcomes colour consistency issues while providing sufficient discrimination to distinguish between different colours.

This paper is organised as follows: Section 2 provides a survey of related work. The system components are described in Section 3. Sections 4 and 5 outline the main parts of the tracker; object detection and object tracking modules. Experimental results are presented in Section 6 along with discussions and possible further improvements. It is followed by a conclusion in Section 7.

2. Related work

The topic of tracking non-rigid objects by appearance has been tackled using various image cues. Colour, motion, shape, depth are the common appearance modalities used in such work. As mentioned in Section 1, most of them designed to deal with relatively large-scale objects.

Birchfield [4] used colour cues and intensity gradients to control a camera's pan and tilt to track a human head around an untextured and static room where the head is modelled as an ellipse. The attributes of the ellipse are the colour contents inside the ellipse and the intensity gradient around the perimeter. The colour histogram and histogram intersection, introduced by Swain [27], was used as a colour model and similarity measure. The colour model consists of a histogram with eight bins for chrominance elements, $(G - R)$ and $(B - G)$ and four bins for the luminance $(R + G + B)$. A matching score which is a combination of gradient and colour cues is presented. Bradski tracked a person's face to form part of a perceptual user interface [7]. The model was built by sampling skin-coloured pixels, converting to the *HSV* colour space and constructing a histogram of hues if their saturation and value are greater than a predefined threshold. The Pfinder system [28] models parts of a person's body with Gaussian distributions called blobs. The co-ordinate of the blob centroid and *YUV* colour components are encoded and used to track constituent body parts. McKenna et al. [19–21] used a mixture of Gaussian distributions to model and track a multi-colour object. At each time step, the model is fitted by semi-parametric learning based on the *responsibility* of each model component and a cross-validation method. The training and validation sets are sampled hue and saturation pairs from pixels with values within a predefined band. *W4* [11] uses relatively complex shape and intensity to track people, identifying poses and separating individuals during occlusion. Koller et al. [17] used contour and camera placement assumptions to track objects during occlusion. The tracking process in this system is by proximity or simple 2D motion models and a similarity measure of object appearance. These systems require a high-quality representation or a reasonably large number of pixels on the targets, for example, at least 250

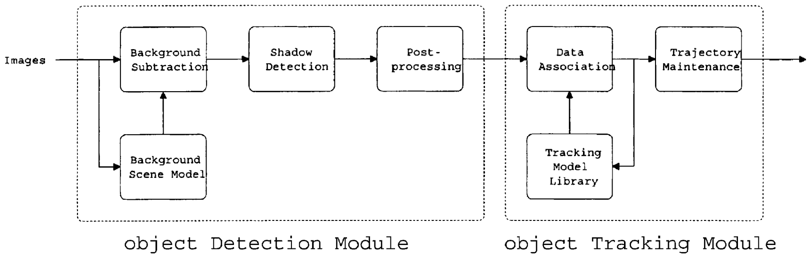

Fig. 2. Outline architecture of our system.

pixels per blob are required for the *W4* approach. On the other hand, systems as in Refs. [10,18,25] employ simple shape information and motion models to classify and track multiple objects. Moving objects are classified into categories such as people or cars. Since the camera has a wide-angle lens and is mounted at a great distance from most targets, the extracted objects in this situation have low resolution and a small number of supporting pixels.

3. System overview

The system consists of two main parts, an object detection module and a target tracking module as show in Fig. 2. The object detection module deals with detecting moving objects from a stationary scene, eliminating shadows and removing spurious objects. The target tracking module takes detected objects from the current frame and matches them to the target models maintained in the target model library. The tracking process is performed in 2D; therefore, a geometric camera calibration is not required. Each target model summaries all past measurements into its appearance and motion models. The target tracking module maintains these target models to facilitate the matching process as well as eliminate spurious trajectories. The outputs of this module are target trajectories that exhibit temporal and spatial consistency.

4. Object detection module

The module for object detection consists of three parts. First, each pixel in the input images is segmented into moving regions by a background subtraction method. The background subtraction uses a per pixel mixture if Gaussians for the reference image to compare with the current image. The outcome is fed into a shadow detection module to eliminate shadows from moving objects. The resulting binary image is then grouped into different objects by the foreground region detection module.

4.1. Background modelling

This section discusses the background model used by our method. The model operates within a similar framework to that introduced by Stauffer and Grimson [10,25]. The difference lies on the update equation of the model parameters and the initial weight of a new Gaussian component (will be explained later in this section). In previous work we have demonstrated the superior performance of update equations derived from sufficient statistics and *L*-recent window formula over other approaches [15,16]. The derivation of the update equations is given in Appendix A. This provides a system which learns a stable background scene faster and more accurately than that of Stauffer and Grimson. Fig. 3 shows

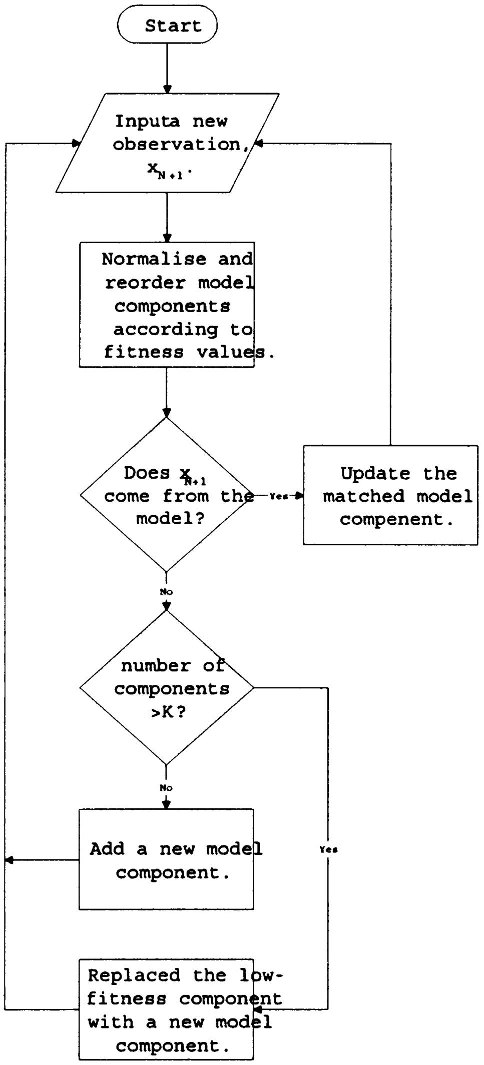
Start
Inputa new observation, x_{N+1}.
Normalise and reorder model components according to fitness values.
Does x_{N+1} come from the model?
Yes
Update the matched model compenent.
No
number of components >K?
No
Yes
Add a new model component.
Replaced the low-fitness component with a new model component.

Fig. 3. Flowchart of background subtraction and maintenance.

a flowchart of the background subtraction and maintenance used in this paper.

In this system, each pixel in the scene is modelled by a mixture of K Gaussian distributions (K is a small number from 3 to 5). Different Gaussians are assumed to represent different colours. The probability that a certain pixel has a value $\mathbf{x}_N$ at frame N, can be written as

$$p(\mathbf{x}_N) = \sum_{j=1}^{K} w_j \eta(\mathbf{x}_N;\ \boldsymbol{\mu}_j,\ \boldsymbol{\Sigma}_j) \tag{1}$$

where w_k is the weight parameter of the kth Gaussian component which represents the time proportions that the colour stays in the scene. Its value also has to satisfy the following stochastic conditions:

$$\sum_{j=1}^{K} w_j = 1 \text{ and } w_k \geq 0;\ \ k = 1, 2, \ldots, K. \tag{2}$$

$\eta(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the Gaussian distribution of kth component or kernel represented by

$$\eta(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{|2\pi\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^{\mathrm{T}} \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right) \tag{3}$$

where $\boldsymbol{\mu}_k$ is the mean and $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}_d$ is the covariance of the kth component. d is the dimensionality of the vector $\mathbf{x} \in \mathbb{R}^d$. This simplification reduces model accuracy but provides a significant increase in efficiency as it removes the need for matrix inversions on a per pixel level.

The background components are determined by assuming the background contains the B highest probable colours. These probable background colours are the ones that remain static for a large portion of time. (Static single-colour objects tend to form tight clusters in the colour space while moving ones form wider clusters due to different reflecting surfaces during movement). A parameter called *fitness* value as w_k/σ_k is introduced to quantify this degree of similarity to be a part of the background for each mixture component. To identify background components, the K distributions are ordered based on the fitness value and the first B distribution is determined by

$$B = \arg\min_b \left(\sum_{j=1}^{b} w_j > th \right). \tag{4}$$

$\{w_1, w_2, \ldots, w_K\}$ are now the weight parameters of the mixture components in descending orders of fitness values. The threshold th is the minimum fraction of the background model. In other words, it is the minimum prior probability that the background is the scene, i.e. $th = 0.8$ is the prior probability that 80% of the scene variation is due to the static background processes.

Background subtraction is performed by marking any pixel that is more than 2.5 standard deviations away from all B distributions as a foreground pixel; otherwise a background pixel.

If the above process identifies any match to the existing model, the first Gaussian component that matches the test value will be updated with the new observation by the update equations,

$$\hat{w}_k^{(N+1)} = \hat{w}_k^{(N)} + \alpha^{(N+1)}(M_k^{(N+1)} - \hat{w}_k^{(N)}) \tag{5a}$$

$$\hat{\boldsymbol{\mu}}_k^{(N+1)} = \hat{\boldsymbol{\mu}}_k^{(N)} + \rho^{(N+1)}(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^{(N)}) \tag{5b}$$

$$\hat{\boldsymbol{\Sigma}}_k^{(N+1)} = \hat{\boldsymbol{\Sigma}}_k^{(N)} + \rho^{(N+1)}((\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^{(N)})(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^{(N)})^{\mathrm{T}} - \hat{\boldsymbol{\Sigma}}_k^{(N)}) \tag{5c}$$

where

$$\alpha^{(N+1)} = \max\left(\frac{1}{N+1}, \frac{1}{L}\right)$$

$$\text{and } \rho^{(N+1)} = \max\left(\frac{1}{\sum_{i=1}^{N+1} M_i^{(N+1)}}, \frac{1}{L}\right).$$

The membership function which attributes new observations to a model component is

$$M_k^{(t+1)} = \begin{cases} 1, & \text{if } w_k \text{ is the first matched Gaussian component;} \\ 0, & \text{otherwise.} \end{cases}$$

Here w_k is the kth Gaussian component, $\alpha^{(N+1)}$ is the *learning rate* and $1/\alpha$ defines the time constant which determines change. N is the number of updates since system initialisation. Eqs. (5a)–(5c) is the approximation of those derived from sufficient statistics L-recent window to reduce computational complexity (see Appendix A).

If no match is found to the existing components, a new component is added. If the maximum number of components has been exceeded the component with lowest fitness value is replaced (and therefore, the number updates to this component is removed from N). The initial weight of this new component is set to $\alpha^{(N+1)}$ and the initial standard deviation is assigned to that of the camera noise.

This update scheme allows the model to adapt to changes in illumination and run in real-time.

4.2. Shadow elimination

In order to identify and remove moving shadows, we need to consider a colour model that can separate chromatic and brightness components. It should also be compatible and make use of our mixture model. This can be done by comparing non-background pixels against the current background components. If the difference in both chromatic and brightness components are within some threshold, the pixel is considered as a moving shadow. We use an effective computational colour model similar to the one proposed by Horprasert et al. [12] to fulfil these needs. It consist of

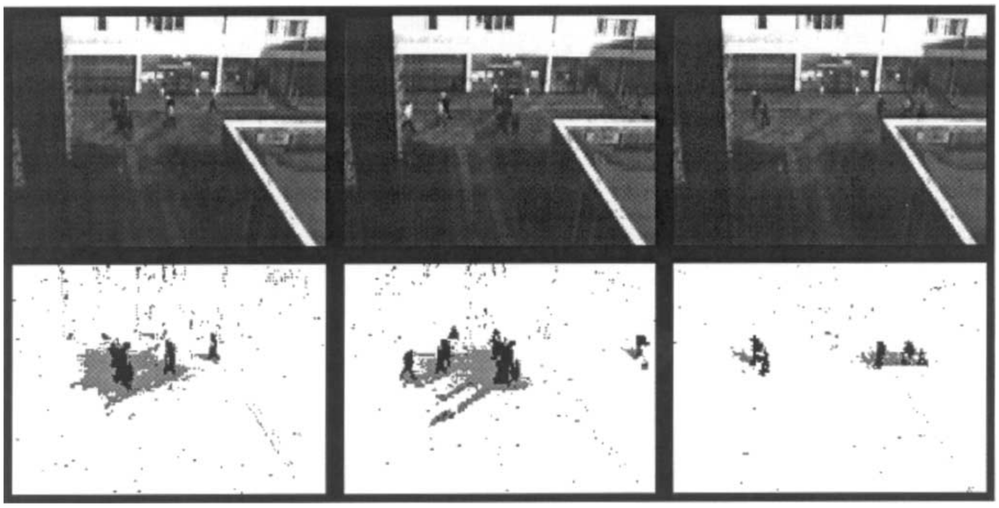

Fig. 4. Background subtraction with shadow suppression. The top row show a sample sequence at different time instances. The second row shows the results from background subtraction with moving shadows detected. Black pixels are deemed parts of foreground objects of interest and the shadows are shown in grey.

a position vector at the *RGB* mean of the pixel background, **E**, an expected chromaticity line, $\|\mathbf{E}\|$, a chromatic distortion, d, and a brightness threshold, τ. For a given observed pixel value, **I**, a brightness distortion, a, and a colour distortion, c, the background model can be calculated by

$$a = \arg\min_z (\mathbf{I} - z\mathbf{E})^2 \tag{6}$$

and

$$c = \|\mathbf{I} - a\mathbf{E}\|. \tag{7}$$

With the assumption of a spherical Gaussian distribution in each mixture component, the standard deviation of the kth component σ_k can be set equal to d, The calculation of a and c are trivial using a vector dot product. A non-background observed sample is considered a moving shadow if a is within, in our case, 2.5 standard deviations and $\tau < c < 1$.

Fig. 4 shows an example of applying our background subtraction with shadow suppression to an outdoor scene. In the sequence, reflection from building glass doors develops long shadows from the foreground objects. It can be seen that most of the moving shadows from the foreground objects. It can be seen that most of the moving shadows can be identified and eliminated from the foreground objects. Another example of our algorithm used in an indoor environment can be found in Ref. [6].

4.3. Postprocessing

Although we have maintained a good model for the background scene, the effects of camera jitter, moving vegetation, camouflage and occlusions cannot be eliminated by background subtraction alone. By analysing the characteristics of these events, we can lessen the problem

- Camera noise normally produces regions with small areas. These can be eliminated by applying morphological operations or simply by discarding objects of a small number of supporting pixels.
- The multi-colour background model can reduce errors resulting from camera jitter, moving vegetation and secularity. However, if some part of the background (a tree branch for example) exhibits an unseen motion (due to a strong gust of wind), a false detection occurs, as there is no part of the model that represents the pixel colour. These problems share similar characteristics to those of the first point and can be discarding objects with small ration of area to boundary length.
- Camouflage and occlusion from stationary objects presents difficulties in segmentation that can not be simply overcome. In our system, the object-tracking module handles these problems.

The binary image from background subtraction and shadow detection still contains noise. It is passed into a size filter to remove small noise artefacts. A connected component analysis is then performed to detect foreground regions. The results from this module are list of detected foreground regions. The results from this module are list of detected foreground objects, their characteristics as well as a noiseless binary image. These results are used by the data association and stochastic sampling search modules.

5. Target tracking module

The target tracking module deals with assigning foreground objects detected from the object detection module to models maintained in the target model library. It also handles situations such as new targets, lost or occluded targets, camouflaged targets and targets whose appearance merges with others. This task incorporates all available information to choose the best hypothesis to match. The following sections describe the target tracking model used in our system. It consists of data association, stochastic sampling search and the trajectory maintenance modules.

5.1. Target model

In our system, multiple objects are tracked based on information of their position, motion, simple shape features and colour contents. The characteristics of an object can be assumed to be uncorrelated, as we place no constraints upon the types of objects that can be tracked. Therefore, separate probabilistic models are employed for motion, simple shape and colour contents. The following subsections explain each model and the method to update the statistics of its state.

5.1.1. Motion model

The co-ordinates of the object's centroid are modelled by a discrete-time kinematic model. Kalman filters are employed to maintain the state of the object. The centroid is modelled by a white noise acceleration model. This is to correspond with the assumptions that objects move with a near constant velocity. The object manoeuvring ability is encoded in the process covariance matrix as described below. The state equation for the centroid **x** co-ordinate is derived from the piecewise-constant white acceleration model [1] by

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{v}(k) \tag{8a}$$

$$\mathbf{v}(k) = \mathbf{M}(k)v(k) \tag{8b}$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix}, \qquad \mathbf{F}(k) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{M}(k) = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}.$$

$\mathbf{x}(k)$, $\mathbf{F}(k)$ and $\mathbf{M}(k)$ are the state, state transition matrix and acceleration gain matrix of the **x** co-ordinate at time instance k. The process noise $v(k)$ is a sequence of zero-mean, white, Gaussian acceleration noise with covariance $\mathbf{Q}(k)$ and Δt represents the time difference between frame $k+1$ and k. The measurement equation for this co-ordinate is written as

$$\mathbf{z}(k+1) = \mathbf{H}(k+1)\mathbf{x}(k+1|k) + \mathbf{w}(k+1) \tag{9}$$

with the measurement matrix, $\mathbf{H}(k) = [1\ 0]$ and $\mathbf{w}(k+1)$ a zero-mean, white, Gaussian noise with measurement covariance $\mathbf{R}(k+1)$. The process noise covariance and measurement variances are

$$\mathbf{Q}(k) = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 \end{bmatrix}\sigma_{\mathrm{p}}^2$$

and

$$\mathbf{R}(k+1) = \sigma_{\mathrm{m}}^2$$

where σ_{p}^2 and σ_{m}^2 represent the uncertainties of acceleration and measurement, respectively. The centroid **y** co-ordinate is modelled in the same way.

Since the object is tracked through outdoor background clutter, the problems of camouflage and occlusion frequently occur. Partial occlusions may provide motion measurements that can be used to update the motion model if they are detected. If no measurement satisfies a tracking model, the state variables can still propagate through time. This results in the spreading of the probability density function of the state variables. To help the data association module reduce the chance of false matching, a model-switching Kalman filter is introduced. It is based on different process noise levels for normal, occluded and lost tracks where

$$\sigma_{\mathrm{p,normal}}^2 \le \sigma_{\mathrm{p,occlused}}^2 \le \sigma_{\mathrm{p,lost}}^2.$$

The **y** co-ordinate centroid is modelled with the same equations.

The track formation is based on 2/2 logic procedure [2] with the initial process noise set to $\sigma_{\mathrm{p,normal}}^2$. More details of this track formation are given in Section 5.4.

These variables can be assumed uncorrelated and the final equations can be obtained by augmenting the vectors blockwise and the matrices block-diagonally. The model is initialised by the two-point differencing method [1] after the formation, as

$$\mathbf{x}(1|1) = \begin{bmatrix} z(1) \\ \dfrac{z(1)-z(0)}{\Delta t} \end{bmatrix}, \text{ and} \tag{10a}$$

$$\mathbf{P}(1|1) = \begin{bmatrix} 1 & \dfrac{1}{\Delta t} \\ \dfrac{1}{\Delta t} & \dfrac{2}{\Delta t^2} \end{bmatrix}\sigma_{\mathrm{m}}^2. \tag{10b}$$

where **P** is the state covariance matrix, $z(0)$ and $z(1)$ are the position measurements obtained at the first and second instances that the object appeared to the field of view respectively.

5.1.2. Shape model

Simple shape information about the object is useful in identifying what type of tracking is applied for each detected foreground region in the current frame. It can be represented by the height and width of the minimum bounding box of the object. The extensive change of

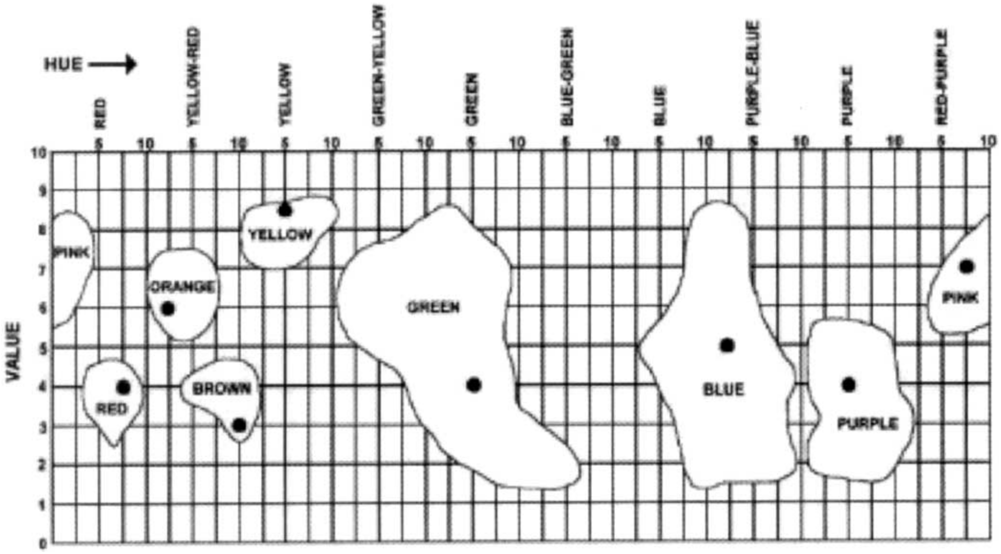

Fig. 5. Location of consensus areas and focal points on a two dimensional representation of the Munsell space.

the shape size from an average size may indicate an object under camouflage or partial occlusion.

The average-height estimate, $\hat{h}$ is maintained as follows (and similarly for width $\hat{w}$).

$$\hat{h}(k+1) = \hat{h}(k) + \beta(k+1)(h(k+1) - \hat{h}(k)) \tag{11}$$

where $\beta(m) = \max(1/L, 1/m)$, $\hat{h}(0) = h(0)$ and $h(k+1)$ is the instantaneous height measurement at frame $k+1$ of the object's life.

5.1.3. Colour model

Most colour tracking systems model multi-coloured objects using a colour histogram or a mixture of Gaussian distributions. If only a small set of samples is available, most of the bins in the histogram are empty and the solution of fitting data to a mixture model tends to converge to a singular solution. One answer to this is to increase the bin size or limit the number of Gaussian components in the mixture. This leads to the question of how large the bin size should be or how many components in the mixture is sufficient to represent the colour distribution of the object. Too large a bin size or too few mixture components can result in no discernible difference between objects. Too small a bin size or too many mixture components may not only cause singular solutions but also be unable to cope with changes resulting from objects passing over different regions in the scene (lighting condition can change from one region of the image to another). Our colour model utilises a colour transformation obtained from consensus colours in Munsell colour space by converting observed colours into eleven basic colours [3]. This consensus colour was experimentally developed by Sturges et al. [26]. Fig. 5 and Table 1[1] display the Sturges' consensus areas and focal points of chromatic and achromatic colours, respectively. The ill-defined regions of the colour space shown in Fig. 5 depict the colours which are ambiguously interpreted by human subjects. The colour conversion for this region is obtained by the nearest neighbour from the colour point to the edge of the consensus area. The colour histogram therefore contains eleven normalised bins. The L-recent instantaneous histograms will be maintained in the model by

$$\hat{B}_i(k+1) = \hat{B}_i(k) + \beta(k+1)(B_i(k+1) - \hat{B}_i(k)) \tag{12}$$

where $\beta(m) = \max(1/L, 1/m)$, $\hat{B}_i(0) = B_i(0)$; $i = 2, \ldots, M$ and $B_i(k+1)$ is the ith bin of the instantaneous histogram at frame $k+1$.

An example of the result from the conversion is shown in Fig. 6(d). In the figure, only foreground pixels are converted into consensus colours via a look-up table. It can be seen from Fig. 6(d) that several pixels do not have colours as would be expected by the human eye. This is because the conversion was performed on the individual pixels rather than the spatial relationship within a neighbourhood. Ambiguous pixels are classified on a nearest neighbour basis. Since the consensus colours were obtained by presenting homogenous colour patches to the subject [26], colour segmentation is poorer than human perception. Improvements could be made by segmentation and

Table 1
Location of consensus areas and focal points of achromatic colours

Colour	Focal point	Consensus range
White	9.5	$V \geq 9$
Grey	5.5	$4 \leq V \leq 7$
Black	0.5	$V \leq 1$

[1] In Table 1, the focal points for white and black are not $V = 10$ and $V = 0$, respectively, as they are not achievable using conventional pigments in the original experiments.

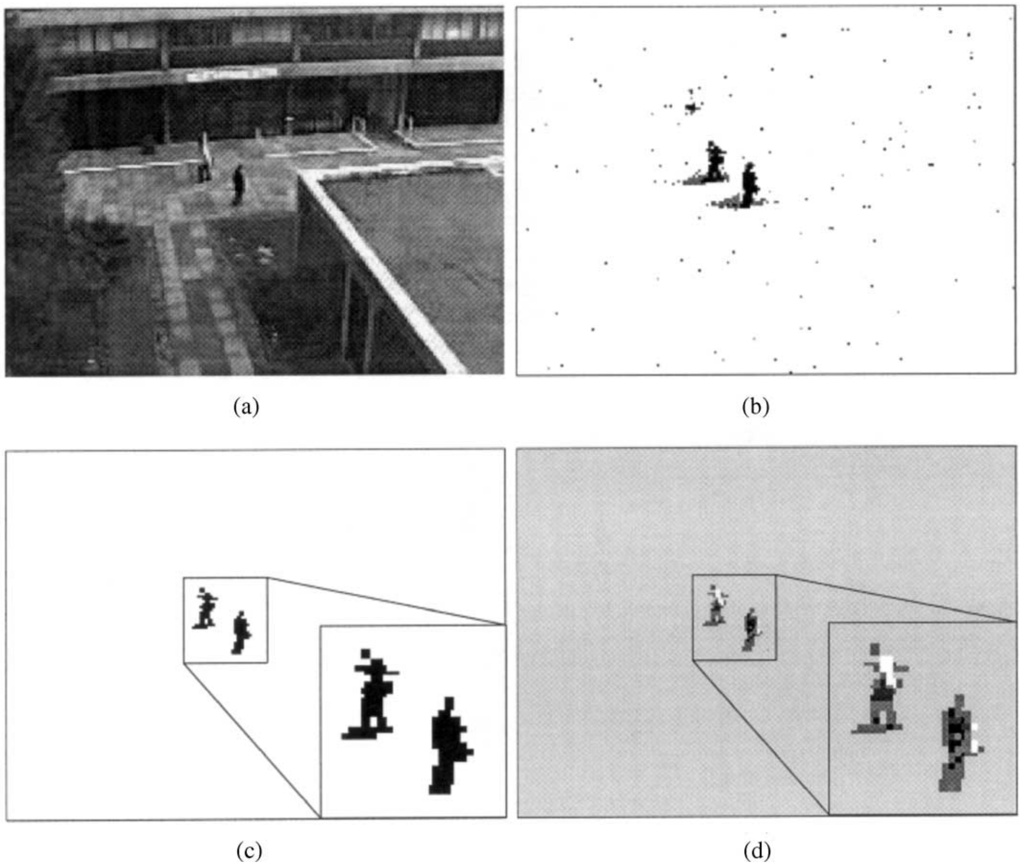

Fig. 6. The consensus colour conversion. (a) the original image, (b) the result from background subtraction and shadow elimination, (c) the foreground region detection, (d) the colour-converted objects over the background image (cyan colour).

classifying the colour of the region according to the majority.

5.2. Data association module

This module makes use of both the motion and appearance models of the targets. Each tracking model includes motion and appearance models. The motion model gives an ellipsoidal prediction area called the *validation gate* [5]. This area is represented by a squared Mahalanobis distance of less than or equal to a *gate threshold* from the predicted measurement with a covariance matrix being the Kalman innovation covariance matrix, **S**. The squared Mahalanobis distance is a chi-squared distributed with the number of degrees of freedom equal to the dimension of the measurement vector. Hence, the probability of finding the measurement in the gate, i.e. having the Mahalanobis distance less than the gate threshold can be obtained from the chi-squared distribution. The *gate probability*, which is the probability of mass that the true measurement will fall in the gate, is obtained from the cumulative probability distribution tables of the chi-square distribution for various values of gate threshold and dimensions of the measurement vector, shown in Table 2. Fig. 7 shows an example of the locations of gates and measurements. As the targets come close to each other, a measurement generated from one target may fall within more than one gate and an optimal assignment sought.

The purpose of data association is to assign the measurements detected in the current frame to target models. Targets whose parts cannot be detected due to a colour spectrum matching that of the background are deemed to be camouflaged. Objects whose appearances merge (in perspective view) normally have a sudden increase in size in either or both horizontal and vertical directions. As camouflaged and partially occluded objects share the same characteristic of having a rapid reduction/growth in size and shape, this can be used to identify its occurrence. These are dealt with using the Stochastic Sampling Search (SSS) in Section 5.3. However, the search

Table 2
Gate probability with various gate thresholds and measurement dimensions

Dimensionality of measurement	Gate threshold			
	1	4	9	16
1	0.68269	0.95450	0.99730	0.99994
2	0.39347	0.86466	0.98889	0.99967
3	0.19875	0.73854	0.97071	0.99887

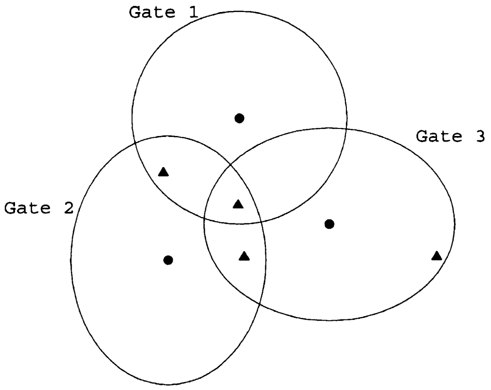

Fig. 7. Example of validation gates (measurements are displayed by ▲'s).

is relatively more computationally expensive and data association can be used to reduce this complexity in normal cases.

The assignment process begins with the calculation of a *validation matrix* (or *hypothesis matrix*) whose rows represent all detected foreground objects and columns all targets in the model library. Observations which are within this gate, will have their appearance similarity calculated. This starts by determining the shape of the object. If the shape does not change extensively, its colour similarity is calculated. If its similarity score exceeds a predefined threshold, the matching score is placed in the validation matrix. The score is based on a combination of model similarities for both motion and appearance, represented by

$$T_{ij} = M_{ij} + H_{ij}. \tag{13}$$

The motion score M_{ij} is represented by

$$M_{ij} = \Pr(Z > z_{ij}) \tag{14}$$

where z_{ij} is the Mahalanobis distance of the ith measurement $(\mathbf{z}_i(k+1))$ to the estimated position predicted from the jth target $(\hat{\mathbf{z}}_j(k+1|k))$.

$$z_{ij} = ((\mathbf{z}_i(k+1) - \hat{\mathbf{z}}_j(k+1|k))^{\mathrm{T}}\mathbf{S}_j(k+1)^{-1}(\mathbf{z}_i(k+1) - \hat{\mathbf{z}}_j(k+1|k)))^{1/2}.$$

$\Pr(Z > z_{ij})$ is the standard Gaussian cumulative probability in the right-hand tail which gives the maximum value of 0.5 if the measurement coincides with the predicted (mean) location. (This can be implemented in a look-up table to increase speed of operation.)

The colour similarity H_{ij} between the object i and the target j is calculated by histogram intersection

$$H_{ij} = \frac{1}{2}\sum_{k=1}^{11}\min(B_{ik}, \hat{B}_{jk}) \tag{15}$$

where $\{B_{i1}, B_{i2}, \ldots, B_{i11}\}$ is the normalised colour histogram of the object i and $\{\hat{B}_{i1}, \hat{B}_{i2}, \ldots, \hat{B}_{i11}\}$ is the normalised colour histogram of the target j.

The total score can be thought of as a combination of two histogram intersections. In the calculation of the motion score, a continuous intersection of two Gaussian distributions, each of the same variance is utilised. However, approximation of the two-dimensional Gaussian intersection by a one-dimensional intersection is made to reduce the computation load. This is done using the Mahalanobis distance to represent the normalised standard deviation in the calculation of the one-dimensional standard Gaussian cumulative probability (as seen in Eq. (14)). Therefore, the total score ranges from 0 to 1 with increasing values representing increasing similarity of a particular object to a target.

The search for the best solution is called the *assignment problem*. The best solution can be defined as the assignments that maximises the hypothesis score which is

$$\varphi = \sum_{(i,j)\in\mathscr{H}} T_{ij} \tag{16}$$

where the hypothesis is represented by an unordered set of measurement-to-track assignments, $\mathscr{H} = \{(trk_1, mea_1), (trk_2, mea_2), \cdots, (trk_r, mea_r)\}$. The tracks and measurements in the hypothesis must not be repeated. One simple solution is to modify the existing validation matrix by adding new hypothesised targets or undetected measurements to form a square matrix and run an assignment algorithm such as the *Hungarian algorithm*[2] [8].

Fig. 8 shows over a sequence of images. To make a better contrast, colour lines and boxes are plotted over a low contrast image as shown in Fig. 8(b). In the figure, the validation gate of the target's centroid of each Kalman filter is the maximum inscribed ellipse of the black box. The trajectories are shown in different grey levels.

After the data association process, all assigned measurements are removed from the binary map. The binary map is then passed to the SSS (described in Section 5.3) to extract measurement residuals available for the unassigned targets.

5.3. Stochastic Sampling Search

If camouflages and occlusions occur, measurements of some observations may not be assigned a target. All unassigned targets are then passed to the SSS along with the binary map obtained from the object detection module with all assigned measurements removed. The SSS is a method that incorporates measurement extraction, motion tracking and data association in the same process.

It begins by sorting the unassigned target models according to **y** co-ordinate. The search for the optimal

[2] Other less expensive algorithms as well as the ones which do not assume square matrix are available. (See Ref. [5] pages 342–346 for more details.)

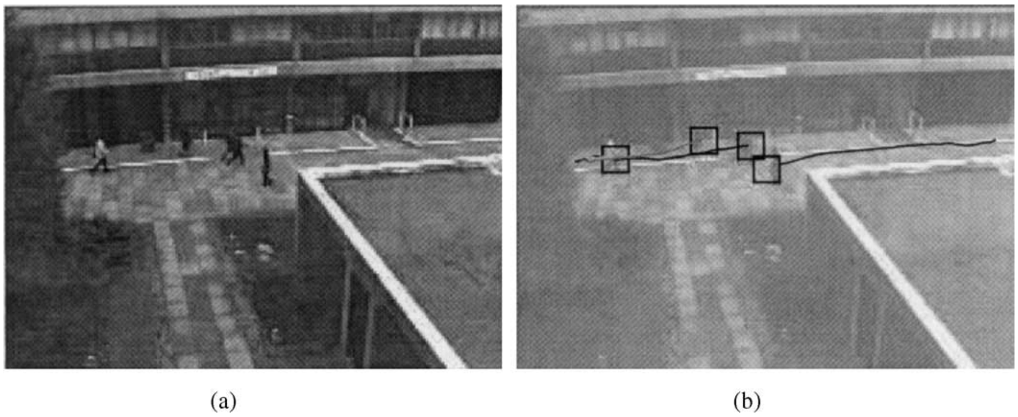

Fig. 8. Multiple target tracking by data association. (a) Original image, (b) reduced contrast image with tracking.

foreground region for each target model starts from the target with maximum value of **y** co-ordinate of the image space.[3] This is based on an assumption that targets move in a ground plane in perspective view. (This assumption may be dropped for general cases and other methods of selecting a target according to depth order can be used.) A number of patches with approximately the same size as the target are generated.[4] The locations of these patches are randomly sampled from the probability density function (pdf) of the motion model. In this case, the two-dimensional Gaussian distribution has a mean at the predicted measurement $\hat{\mathbf{z}}_j(k+1|k)$ and a covariance matrix being the innovation covariance matrix $\mathbf{S}_j(k+1)$. In each patch, only the pixels marked by the binary image are considered and converted to consensus colours and a colour histogram derived from the result. However, this patch normally includes pixels generated from other targets. Normal histogram intersection would not give optimal results. Instead of normalising the histogram, each bin in the histogram is divided by the estimated number of pixels in the target before the colour similarity is calculated. The patch i generated from target j is represented as

$$P_j^i = \{\mathbf{z}_j^i, B_{i1}, B_{i2}, \ldots, B_{i11}\} \tag{17}$$

with

$$\mathbf{z}_j^i \sim \eta(\hat{\mathbf{z}}_j(k+1)|k), \mathbf{S}_j(k+1)). \tag{18}$$

$\mathbf{z}_j^i$ is the centre of the patch and $\{B_{i1}, B_{i2}, \ldots, B_{i11}\}$ are bins 1–11 of the unnormalised histogram. The colour similarity can then be calculated by

$$H_j(P_j^i) = \frac{1}{2}\sum_{k=1}^{11}\min\left(\frac{B_{ik}}{\hat{N}_j}, \hat{B}_{jk}\right) \tag{19}$$

where $\hat{N}_j$ is the current estimated number of pixels of the target j.

The motion score of each patch is also calculated from the motion model by

$$M_j(P_j^i) = \Pr(Z > z_j^i) \tag{20}$$

where z_j^i is the Mahalanobis distance of the ith patch ($\mathbf{z}_j^i$) to the predicted position from the jth target ($\hat{\mathbf{z}}_j(k+1|k)$), written as

$$z_j^i = ((\mathbf{z}_j^i - \hat{\mathbf{z}}_j(k+1|k))^{\mathrm{T}}\mathbf{S}_j(k+1)^{-1}(\mathbf{z}_j^i - \hat{\mathbf{z}}_j(k+1|k)))^{1/2}.$$

The model similarity of the patch and the target based on both motion and appearance models can be calculated by

$$T_j^i = M_j(P_j^i) + H_j(P_j^i). \tag{21}$$

This score is similar to the one found in Section 5.2 which is based on the model similarity measures. One point worth noting is that the colour similarity, calculated by Eq. (19), unlike Eq. (15) does not possess scale-invariance. This is acceptable, as the occlusions or camouflages are relatively short compared to the change in size of the targets over time.

The optimum patch can be obtained as

$$n = \arg\max_i T_j^i \tag{22}$$

and can be used to update the motion model (however, not the appearance model) provided that the similarity of the patch exceeds a predefined threshold. This threshold is the approximate percentage of the visible area of the target. The acceleration noise variance of the matched models from this module is set to $\sigma_{\mathrm{p,occlused}}$ and the unmatched to $\sigma^2_{\mathrm{p,lost}}$.

At the end of each target search, the matched optimum patch will have its contents removed from the binary map.[5] The binary map is then passed to the track maintenance module to identify new targets.

[3] The image space has its origin at the top corner and positive axes point to the right and down directions.

[4] In our experiments, patches with a pixel in each larger than the average size of the target gave the best results.

[5] The SSS cannot be used to track any target at the beginning of its appearance. It only operates after a visual model of the target has been established.

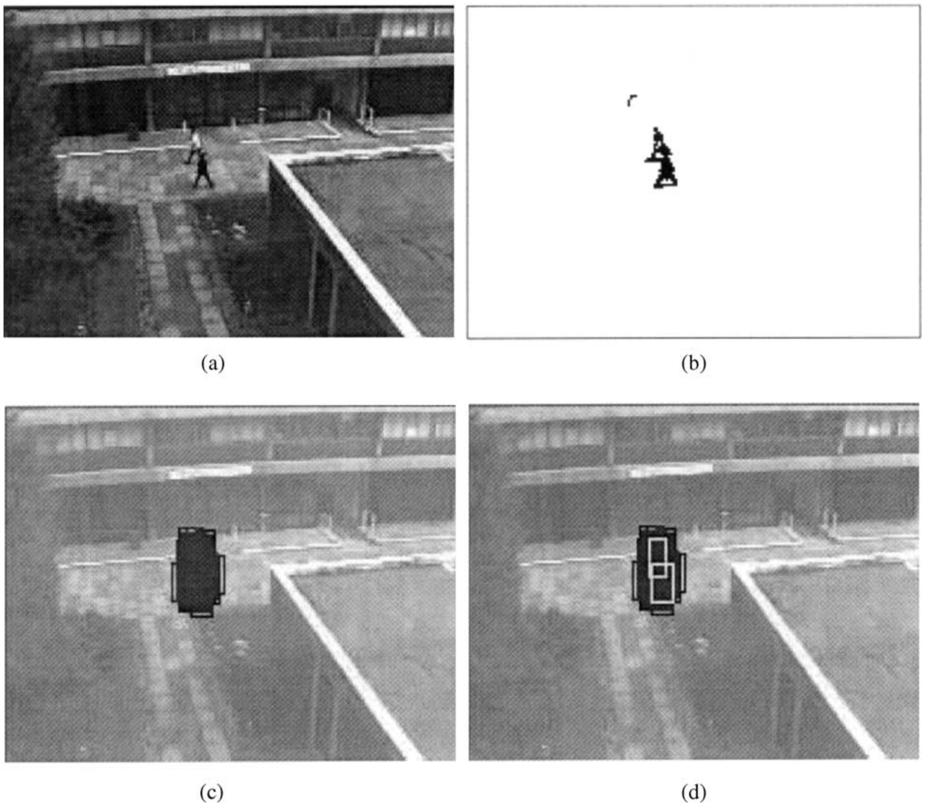

Fig. 9. The Stochastic Sampling Search. (a) The original image, (b) the binary map from the foreground region detection, (c) the patches generated by the Stochastic Sampling Search, (d) the optimal patches selected.

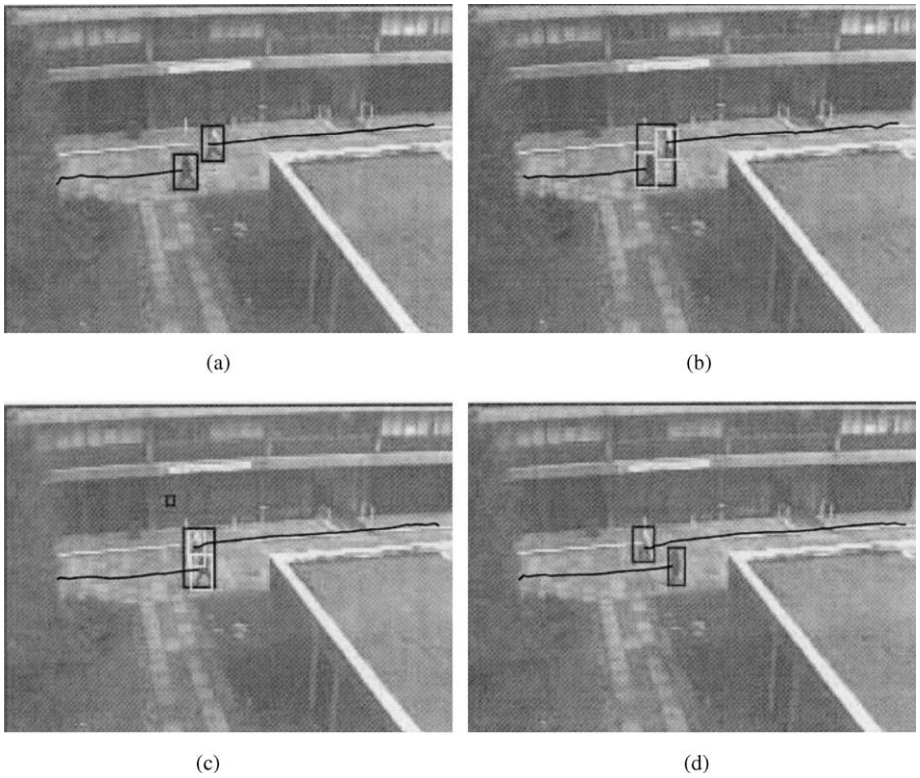

Fig. 10. Tracking through occlusion. (a) The targets are being tracked by tracking and data association, (b)–(c) tracking by Stochastic Sampling Search, (d) the tracking and data association resumes its operation.

Figs. 9 and 10 show the process of SSS and the result of tracking through occlusion, respectively.

In Fig. 9(c), a number of patches were randomly generated from a two dimensional Gaussian distribution with the mean and covariance obtained from the Kalman prediction. The patch with the maximum score for each object was selected as a measurement for the correction step of the Kalman filter and shown with white boxes.

Fig. 10 shows an example of tracking targets through occlusion. Each target was tracked by the tracking and data association before the occlusions occur as shown in Fig. 10(a). The bouncing boxes are the extents of the targets. In Fig. 10(b), the image of both targets merged into a single large object. Since the shapes of both targets changed significantly, this object was not assigned to any track. (However, if the shapes do not change sufficiently, the data association will identify that the number of competing targets is more than the number of objects and assignment will be left for the SSS.) The objects were extracted by the SSS and are shown with white boxes. As the object dynamics of each target was maintained, the data association resumed its operation after the occlusion.

5.4. Trajectory maintenance module

The track maintenance module is designed to deal with trajectory formation, trajectory deletion as well as to eliminate spurious trajectories that occur from unpredicted situations in outdoor scenes such as trajectories resulting from noise and small repetitive motions. By performing connected component analysis on the residual binary image, a list of new objects which have a suitable number of pixels is extracted. This provides evidence of all objects not already accounted for by the tracking system. As the system is running in real time, an object's appearance may not be fully visible within the first few frames of its introduction. The formation of the object's appearance model can be deferred using some specific number of frames or some heuristics about the scene (such as objects at the edge of the camera view).

Track formation is based on 2/2 logic [1]. The process is described as follows. First, every unassigned measurement is used to form a track, called a *tentative track*. At the next frame, a gate is formed by propagating the process and measurement uncertainties from the last position of the target. If a measurement is detected in the gate, this tentative track becomes a normal track; otherwise this tentative track is discarded. The construction of an appearance model for the new target is deferred (as the object's appearance is normally unstable.) For example, a person who has just come out from the building is composed of not only his body but also the door as it is swinging. The tracking process during this period relies solely on the motion model.

If no measurement has been assigned to a *normal track*, it is then changed to a *lost track*. If a normal track is not assigned a measurement during the data association process, it is changed to an *occluded track*. Any type of track can be changed back to normal track if it is assigned a measurement during the data association process. Tracks that have been lost for a certain number of frames or time period T_{lost} will be deleted. This also applies to occluded tracks with T_{occlude} as the allowable number of occlusions before terminating the track. Both T_{lost} and T_{occlude} are used to reduce the chance of drifting and mismatching during and after occlusion. Fig. 11 depicts the process as a state diagram.

Fig. 12 shows the tracking of multiple independent targets moving within the scene. Some measurements are assigned to their targets by data association while others are

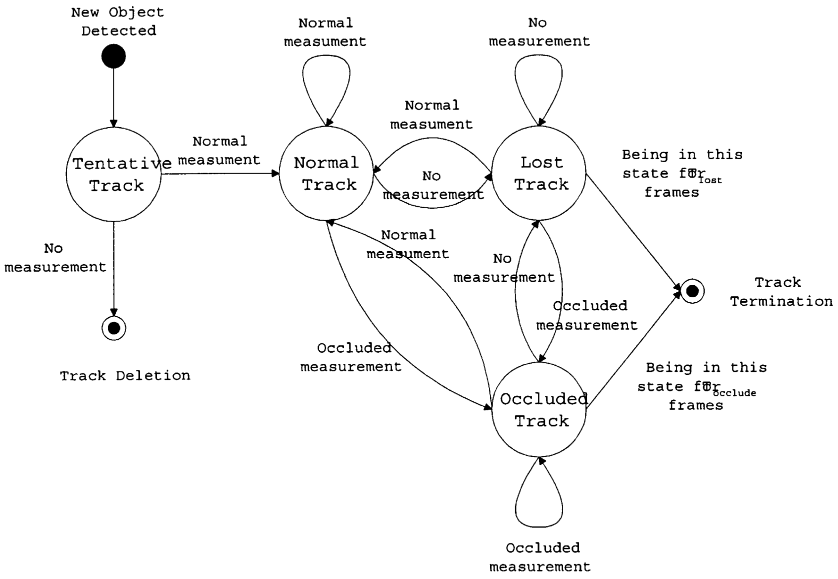

Fig. 11. State diagram of track maintenance.

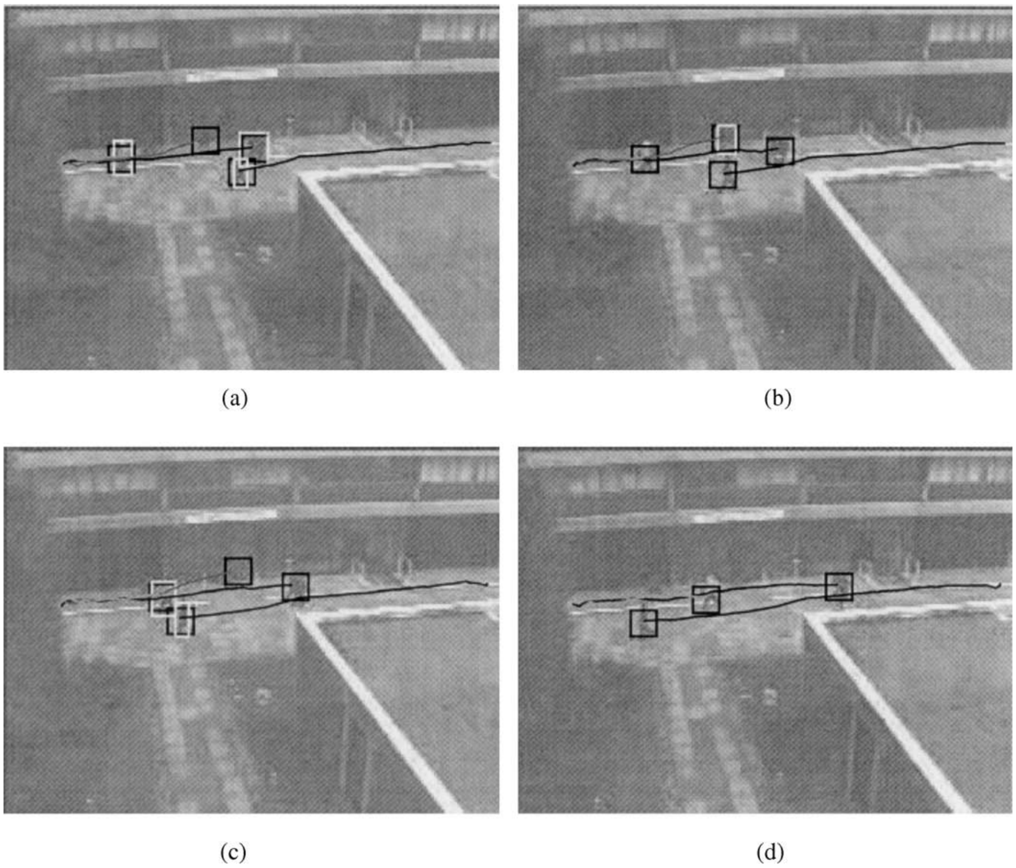

Fig. 12. Tracking multiple independent targets. (a)–(d) tracking with a combination of the data association and Stochastic Sampling Search.

assigned by the SSS as occlusions and camouflages occurring from time to time. The search areas are identified by black boxes while the white boxes indicate the optimal patch from the SSS. Trails are drawn with different grey levels. It can seen that a group of people are tracked as a single object. This is due to all members of the group appearing at the same time when they entered the scene. Some close objects are identified as a single large object due to imperfections in shadow elimination. In Fig. 12(c), one of the targets entered the building; however, the search continues for a certain period before being terminated.

6. Experimental results

Some results of applying the proposed technique for tracking low-resolution targets are presented in this section. Fig. 13(a) and (b) show the cumulative trajectories of objects tracked during one hour of operation from two cameras. The cameras were set to monitor two different locations of the campus. The tracker runs on a PC Pentium III 450 MHz at approximately 5 frames per second. It is evident that many spurious trajectories are constructed during the period. This is mainly due to insignificant movements of vegetation and occlusions/deocclusions of the clouds in the scenes.

Table 3 summarises the results of tracking through occlusions from the experiment. It can be seen that the SSS performed sufficiently well. However, as limited by T_{lost} and T_{occlude}, tracking over too an long occlusion terminates prematurely. This is caused by people stopping moving and engaging in long conversations where parts of them have been incorporated into the background model. The main problem for the algorithm is group interaction. This includes the case that silhouettes of individual members are joined at the beginning of their appearance then they separate due to different speeds or directions. To identify individual members of unknown classes in a group requires knowledge-based or computationally complex algorithms; however, these mechanisms have not been addressed in this work.

The idea of using a guided random search in visual tracking is utilised by many researchers. The *Con*ditional *den*sity propag*ation* (CONDENSATION) algorithm or particle filter [13,14] and Joint Likelihood Filter (JLF) [22,23] are examples of guided random search approaches in computer vision. The main application of these algorithms is in the area of large-target tracking where features on the targets are reliable. However, background clutter plays an important role in distracting the tracker. The primary goals of using a random search are to cope with rapid movement, occlusion of targets due to static objects or

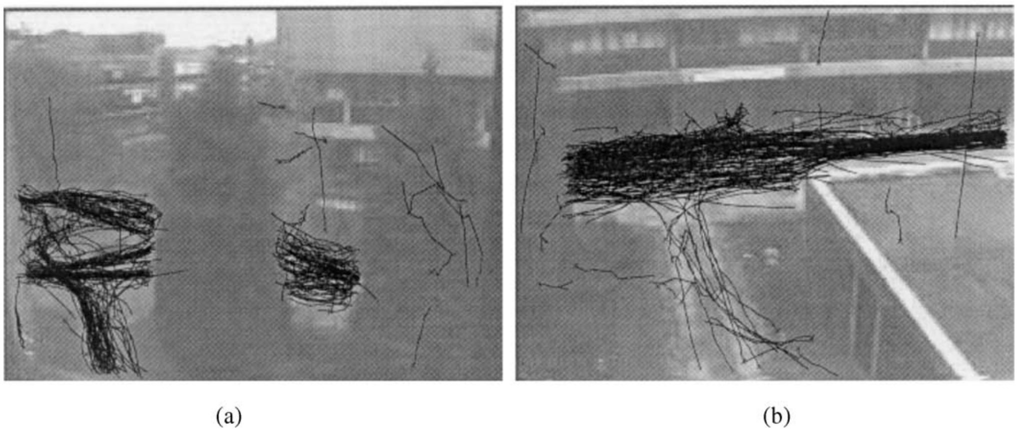

Fig. 13. Tracking results from two cameras. (a) trajectories of objects moving over an hour period in the first camera (with short-lived trajectories removed), (b) trajectories obtained from the second camera.

other targets and distraction by background clutter. The search provides a mechanism of jumping out of local minima in the image space to hopefully find a global solution which corresponds to the correct target assignment. While in CONDENSATION and JLF, the motion information is implicitly incorporated in the search, explicit inclusion of this information is made in our system. The reason behind this is that colour information obtained from small objects is not a reliable as for that of large objects. On the other hand, distraction is small in our system since only foreground pixels are considered in the process. Another advantage of using a random search is that is constrains the classification between targets to local regions. Unlike methods based only on pixel classification of the merged regions [19] which classifies pixels into targets using the posterior probability of the pixels corresponding to the targets before the merge, pixels which exceed the possibility of being a single target will never be assigned to the target being examined.

Table 4 summarises the results of tracking through the field of view of each camera. Number of successes are counted only for the targets that can be tracked throughout the field of view. It can be seen that most of errors are due to camouflage which was caused by the imperfection of the foreground object identification. The effect was severe in the sequence from Camera1. This was caused by the perspective view in that camera. The objects further away from the camera were very small and camouflage made the objects smaller so that they could not pass the size filter and could not be detected from the start of their appearances. Although, we could set the size filter to pass in smaller objects, it could generate a larger number of false detections due to insignificant movements from the trees (as two sizable trees were close to the camera). The second camera does not suffer from perspective size; in contrast, the two large black walls of the building in the scene do cause camouflage when people wearing black clothes walk close to them. Target stopping moving is also a problem inherited from the background subtraction. Even though the tracker can deal with most kinds of movements in the scene, agile movements exhibited by few targets could not be handled. This is due to the low-order kinematic model and the low levels of noise variances used in our tracker.

Table 3
Results of tracking through occlusions from two locations over one hour period

Items	Camera1		Camera2	
	Number of objects	%	Number of objects	%
Total number of occlusions	60	100	81	100
Successes	51	85.00	65	80.25
Failures due to long occlusions	3	5.00	4	4.94
Failures due to group interactions	6	10.00	12	14.81

Table 4
Results of tracking through the field of views in two cameras over one hour period

Items	Camera1		Camera2	
	Number of objects	%	Number of objects	%
Objects in the field of view	407	100	198	100
Successfully tracked through the view	320	78.63	153	77.27
Failures due to camouflage	66	16.22	18	9.09
Failures due to targets stop moving	11	2.70	9	4.55
Failures due to group interactions	6	1.47	12	6.06
Failures due to agile movements	4	0.98	6	3.03

7. Conclusions

We have presented a method to track low-resolution moving objects mainly for outdoor surveillance applications, using colour, simple shape and motion information. The key strength of this method is the use of robust background modelling, the colour mapping used and a novel guided random search. The update equations used in the background subtraction method provide a model which can adapt to scene content and quickly converges upon a stable reference image. The addition of shadow suppression greatly enhances the performance of the basic tracker. The use of the perceptualised colour model to model low-resolution colour targets provides robustness to small objects with minimal colour information and overcomes problems with colour consistency and lighting variation. To use this we have proposed an effective probabilistic measure that combines motion and appearance information. We also introduce a scheme which utilises different degrees of computational complexity dependent on the reliability of the information obtained from the measurement extraction process. Due to the robust background modelling, few false detections are produced. However, some errors in segmentation, for instance camouflages, are inherent to the background subtraction method. These limitations can be overcome by modelling both the static scene and the motion and appearance of the foreground object. A data association process is used in less ambiguous cases. Occlusions are dealt with in much the same way as camouflage as they share some characteristics. This is done by a random search algorithm which utilises both motion and colour information of the target models to track targets through occlusions and camouflage. We have demonstrated the technique on image sequences of unconstrained external environments and have successfully operated the tracker for long periods with no user intervention where the adaptive nature of the approach is capable of overcoming changes in time of day, weather, seasons and activity.

Appendix A. Stochastic on-line approximation

This appendix presents a set of update equations based upon the batch EM algorithm for learning a Gaussian mixture model [9]. From these, a number of incremental (on-line) EM algorithms are proposed.

A.1. Standard batch EM algorithm

The standard EM algorithm is an iterative optimisation algorithm for maximising the likelihood function of several probabilistic models such as Gaussian mixture models and Hidden Markov Models. It gains popularity among researchers due to its simplicity, efficiency and the ability to handle the stochastic constraints (Eq. (2)) naturally. In the case of the Gaussian mixture model, the EM algorithm comprises of two steps and is described as follows.

Expectation:

$$p(w_k|\mathbf{x}_i) = \frac{\hat{w}_k^{[t-1]}\eta(\mathbf{x};\hat{\boldsymbol{\mu}}_k^{[t-1]},\hat{\boldsymbol{\Sigma}}_k^{[t-1]})}{\sum_{j=1}^{K}\hat{w}_j^{[t-1]}\eta(\mathbf{x};\hat{\boldsymbol{\mu}}_j^{[t-1]},\hat{\boldsymbol{\Sigma}}_j^{[t-1]})}. \tag{A.1}$$

Maximisation:

$$\hat{w}_k^{[t]} = \frac{r_k^{[t]}}{N} \tag{A.2a}$$

$$\hat{\boldsymbol{\mu}}_k^{[t]} = \frac{\sum_{i=1}^{N} p(w_k|\mathbf{x}_i)\mathbf{x}_i}{r_k^{[t]}} \tag{A.2b}$$

$$\hat{\boldsymbol{\Sigma}}_k^{[t]} = \frac{\sum_{i=1}^{N} p(w_k|\mathbf{x}_i)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{[t]})(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{[t]})^{\mathrm{T}}}{r_k^{[t]}} \tag{A.2c}$$

where

$$r_k^{[t]} = \sum_{i=1}^{N} p(w_k|\mathrm{x}_i).$$

$\hat{w}_k^{[t]}$, $\hat{\boldsymbol{\mu}}_k^{[t]}$ and $\hat{\boldsymbol{\Sigma}}_k^{[t]}$ are the estimates of weight, mean and covariance of the kth component at iteration t, respectively. $p(w_k|\mathbf{x}_i)$ is the posterior probability that $\mathbf{x}_i$ is generated from the kth component. The data set $\mathscr{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is assumed to be available before the optimisation process.

The Eqs. (A.2b) and (A.2c) can be rewritten in a shorthand notation as

$$\hat{\boldsymbol{\xi}}_k = \frac{\sum_{i=1}^{N} p(w_k|\mathbf{x}_i)\boldsymbol{\zeta}_k(\mathbf{x}_i)}{\sum_{i=1}^{N} p(w_k|\mathbf{x}_i)}. \tag{A.3}$$

In the case of $\hat{\boldsymbol{\mu}}_k$,

$$\boldsymbol{\zeta}_k(\mathbf{x}_i) = \mathbf{x}_i.$$

In the case of $\hat{\boldsymbol{\Sigma}}_k$,

$$\boldsymbol{\zeta}_k(\mathbf{x}_i) = (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}}.$$

A.2. Incremental EM algorithms

Unlike the batch algorithm, incremental algorithms process data from a sequential stream. The incremental EM algorithm based on expected sufficient statistics can be derived as follows

$$\begin{aligned}\hat{w}_k^{(N+1)} &= \frac{1}{N+1}\sum_{i=1}^{N+1} p(w_k|\mathbf{x}_i)\\ &= \frac{1}{N+1}\left(\sum_{i=1}^{N} p(w_k|\mathbf{x}_i) + p(w_k|\mathbf{x}_{N+1})\right)\\ &= \hat{w}_k^{(N)} + \frac{1}{N+1}(p(w_k|\mathbf{x}_{N+1}) - \hat{w}_k^{(N)}). \qquad \text{(A.4)}\end{aligned}$$

Here, $\hat{w}_k^{(N+1)}$ represents the weight parameter estimate at the data instance $N+1$.

Similarly,

$$\hat{\boldsymbol{\xi}}_k^{(N+1)} = \hat{\boldsymbol{\xi}}_k^{(N)} + \frac{p(w_k|\mathbf{x}_{N+1})}{\sum_{i=1}^{N+1} p(w_k|\mathbf{x}_i)}(\boldsymbol{\zeta}_k(\mathbf{x}_{N+1}) - \hat{\boldsymbol{\xi}}_k^{(N)}). \qquad \text{(A.5)}$$

The incremental EM algorithm based upon an L-recent window can be derived as follows.

$$\hat{w}_k^{(N)} = \frac{1}{L}\sum_{i=N=L+1}^{N} p(w_k|\mathbf{x}_i).$$

At frame $N+1$,

$$\hat{w}_k^{(N+1)} = \hat{w}_k^{(N)} + \frac{1}{L}(p(w_k|\mathbf{x}_{N+1}) - p(w_k|\mathbf{x}_{N-L+1})). \qquad \text{(A.6)}$$

For $L \gg 1$, $p(w_k|\mathbf{x}_{N-L+1})$ can be approximated by $\hat{w}_k^{(N)}$, therefore we have

$$\hat{w}_k^{(N+1)} = \hat{w}_k^{(N)} + \frac{1}{L}(p(w_k|\mathbf{x}_{N+1}) - \hat{w}_k^{(N)}) \qquad \text{(A.7)}$$

and

$$\begin{aligned}\hat{\boldsymbol{\xi}}_k^{(N+1)} &= \frac{\sum_{i=N-L+1}^{N} p(w_k|\mathbf{x}_i)}{\sum_{i=N-L+2}^{N} p(w_k|\mathbf{x}_i)}\hat{\boldsymbol{\xi}}_k^{(N)}\\ &\quad + \frac{p(w_k|\mathbf{x}_{N+1})\boldsymbol{\zeta}_k(\mathbf{x}_{N+1}) - p(w_k|\mathbf{x}_{N-L+1})\boldsymbol{\zeta}_k(\mathbf{x}_{N-L+1})}{\sum_{i=N-L+2}^{N} p(w_k|\mathbf{x}_i)}. \qquad \text{(A.8)}\end{aligned}$$

In the same way as $\hat{w}_k^{(N+1)}$, we approximate $p(w_k|\mathbf{x}_{N-L+1})$ by $\hat{w}_k^{(N)}$ and $\boldsymbol{\zeta}_k(\mathbf{x}_{N-L+1})$ by $\hat{\boldsymbol{\xi}}_k^{(N)}$. Eq. (A.8) now becomes

$$\begin{aligned}\hat{\boldsymbol{\xi}}_k^{(N+1)} &= \frac{\hat{w}_k^{(N)}}{\hat{w}_k^{(N+1)}}\hat{\boldsymbol{\xi}}_k^{(N)} + \frac{1}{L\hat{w}_k^{(N+1)}}(p(w_k|\mathbf{x}_{N+1})\boldsymbol{\zeta}_k(\mathbf{x}_{N+1})\\ &\quad - \hat{w}_k^{(N)}\hat{\boldsymbol{\xi}}_k^{(N)}) \qquad \text{(A.9)}\end{aligned}$$

Further approximations can be made, for example $\hat{w}_k^{(N)} \approx \hat{w}_k^{(N+1)}$; however, as the occurrence of objects may be sequential, it may be possible to obtain an L-size sequence generated from only a few mixture components. In this situation the resulting weight parameters of the other components can become very small. Therefore, the ratios of weight parameters at frame N to those at frame $N+1$ can be significant.

A.3. Proposed update scheme

To maintain the accuracy of the estimated parameters when the learning process starts and to allow slow changes in the underlying probability density function, we utilise the Eqs. (A.4) and (A.5), derived from expected sufficient statistics in the first L frames and the Eqs. (A.7) and (A.9) from an L-recent window after that. In many cases where the parameters of the mixture do not change too fast, a simple exponentially decaying form can be applied for the equations from an L-recent window. This provides replacements of equations derived from both methods as

$$\hat{w}_k^{(N+1)} = \hat{w}_k^{(N)} + \alpha^{(N+1)}(p(w_k|\mathbf{x}_{N+1}) - \hat{w}_k^{(N)}) \qquad \text{(A.10)}$$

$$\hat{\boldsymbol{\xi}}_k^{(N+1)} = \hat{\boldsymbol{\xi}}_k^{(N)} + \rho^{(N+1)}(\boldsymbol{\zeta}_k(\mathbf{x}_{N+1}) - \hat{\boldsymbol{\xi}}_k^{(N)}) \qquad \text{(A.11)}$$

where

$$\alpha^{(N+1)} = \max\left(\frac{1}{N+1}, \frac{1}{L}\right)$$

and $\rho^{(N+1)} = \max\left(\dfrac{p(w_k|\mathbf{x}_{N+1})}{\sum_{i=1}^{N+1} p(w_k|\mathbf{x}_i)}, \dfrac{1}{L}\right).$

This has an added advantage that it reduces the computational complexity of the algorithm. In the case of updating only the matched component as applied in our tracker, the update scheme is given in Eqs. (5a)–(5c).

References

[1] Y. Bar-Shalom, X.R. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, 1993.

[2] Y. Bar-Shalom, X.R. Li, Multitarget–Multisensor Tracking: Principles and Techniques, Yaakov Bar-Shalom, 1995.

[3] B. Berlin, P. Kay, Basic Color Terms: Their Universality and Evolution, University of California, 1991.

[4] S.T. Birchfield, Elliptical head tracking using intensity gradients and color histograms, CVPR98 (1998) 232–237.

[5] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Boston, MA, 1999.

[6] R. Bowden, P. KaewTraKulPong, M. Lewin, Jeremiah: the face of computer vision, Smart Graphics'02, Second International Symposium on Smart Graphics, ACM International Conference Proceedings Series, Hawthorn, NY, USA, 2002, pp. 124–128.

[7] G.R. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technology Journal (Q2) (1998) Online Journal.

[8] G. Carpaneto, P. Toth, Solution of the assignment problem, ACM Transactions on Mathematical Software 6 (1) (1980) 104–111.

[9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum-likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society Series B 39 (1977) 1–38.

[10] W.E.L. Grimson, L. Lee, R. Romano, C. Stauffer, Using adaptive tracking to classify and monitor activities in a site, CVPR98 (1998) 22–31.

[11] I. Haritaoglu, D. Harwood, L.S. Davis, W4: real-time surveillance of people and their activities, PAMI 22 (8) (2000) 809–830.

[12] T. Horprasert, D. Harwood, L.S. Davis, A statistical approach for real-time robust background subtraction and shadow detection, Frame-Rate99 Workshop (1999).

[13] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, ECCV96 I (1996) 343–356.

[14] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, IJCV 29 (1) (1998) 5–28.

[15] P. KaewTraKulPong, R. Bowden, Adaptive visual system for tracking low resolution colour targets, BMVC01 (2001).

[16] P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, In Second European Workshop on Advanced Video-based Surveillance Systems (AVBS2001), 2001.

[17] D. Koller, J.W. Weber, J. Malik, Robust multiple car tracking with occlusion reasoning, ECCV94 A (1994) 189–196.

[18] A.J. Lipton, H. Fujiyoshi, R.S. Patil, Moving target classification and tracking from real time video, WACV98 (1998) 8–14.

[19] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, Tracking groups of people, CVIU 80 (1) (2000) 42–56.

[20] S.J. McKenna, Y. Raja, S. Gong, Tracking colour objects using adaptive mixture models, IVC 17 (3–4) (1999) 223–229.

[21] Y. Raja, S.J. McKenna, S. Gong, Colour model selection and adaptation in dynamic scenes, ECCV98 (1998) 460–474.

[22] C. Rasmussen, G.D. Hager, Joint probabilistic techniques for tracking multipart objects, In CVPR98 (1998) 16–21.

[23] C. Rasmussen, G.D. Hager, Probabilistic data association methods for tracking complex visual objects, PAMI 23 (6) (2001) 560–576.

[24] M. Seaborn, L. Hepplewhite, J. Stonham, Fuzzy colour category map for content based image retrieval, BMVC99 (1999) 103–112.

[25] C. Stauffer, W.E.L. Grimson, Learning patterns of activity using real-time tracking, PAMI 22 (8) (2000) 747–757.

[26] J. Sturges, W.A. Whitfield, Locating basic colours in the Munsell space, Color Research and Application 20 (6) (1995) 364–376.

[27] M.J. Swain, D.H. Ballard, Color indexing, IJCV 7 (1) (1991) 11–32.

[28] C.R. Wren, A. Azarbayejani, T.J. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, PAMI 19 (7) (1997) 780–785.