Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 24 (2006) 1244–1255
image AND vision COMPUTING
www.elsevier.com/locate/imavis

Object tracking in an outdoor environment using fusion of features and cameras

Quming Zhou[a,*], J.K. Aggarwal[b]

[a]Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA
[b]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA

Received 7 April 2003; received in revised form 6 June 2005; accepted 7 June 2005

Abstract

This paper presents methods for tracking moving objects in an outdoor environment. A robust tracking is achieved using feature fusion and multiple cameras. The proposed method integrates spatial position, shape and color information to track object blobs. The trajectories obtained from individual cameras are incorporated by an extended Kalman filter (EKF) to resolve object occlusion. Our results show that integrating simple features makes the tracking effective and that EKF improves the tracking accuracy when long-term or temporary occlusion occurs. The tracked objects are successfully classified into three categories: single person, people group, or vehicle.

Keywords: Tracking; Classification; Extended Kalman filter; Data fusion

1. Introduction

The efficient tracking and classification of multiple moving objects is a challenging and important task in computer vision. It has applications in surveillance, video communication and human–computer interaction. Recently, a significant number of tracking systems have been proposed. Some address low-level feature tracking while others deal with high-level description, such as recognition, event detection, and trajectory interpretation. The success of high-level description relies on accurate detection and tracking of moving objects and on the relationship of their trajectories to the background. There is a considerable diversity among the trackers proposed by various researchers [1–4]. The most widely used cues in object tracking are spatial position, shape, color, intensity and motion. Many uncontrollable factors such as lighting, weather, unexpected intruders or occlusion may affect the efficiency of tracking when these cues are used in an outdoor environment. One solution is to combine two or more cues. Another solution is to use multiple cameras. We shall consider both of these options in this paper.

1.1. Feature integration

Multi-feature integration has been exploited extensively in featured-based tracking applications. Shi and Tomasi [5] classified a tracked feature as reliable or unreliable according to the residual of the match between the associated image region in the first and current frames. This work [5] was extended by Tommasini et al. [6], who introduced a simple, efficient, mode-free outlier rejection rule for rejecting spurious features. Kaneko and Hori [7] proposed a feature selection method based on the upper bound of the average template matching error. More complex integration methods using statistical models have been presented in [8,9].

Triesch and Malsburg [10] presented a system for integrating features in a self-organized manner. In their system, all features agreed on a result, and each feature adapted to the result agreed upon. Cai and Aggarwal [2] used a Bayesian classifier to find the most likely match of the subject in the next frame. Multiple features were modeled by a joint Gaussian function. Rigoll et al. [11] combined two stochastic modeling techniques. Pseudo-2D Hidden Markov models were used to capture the shape of

* Corresponding author. Tel.: +1 713 4081374; fax: +1 713 3485686.
E-mail address: quming@rice.edu (Q. Zhou).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.06.008

a person within an image frame. A Kalman filtering algorithm was applied to the output of the first model to track the person by estimating a bounding box trajectory indicating the location of the person within the entire video sequence.

1.2. Multi-camera tracking

Single camera tracking is hampered by the camera's limited field of view. Occlusion is always a challenge for single camera trackers, while multi-camera trackers can utilize the different views to obtain more robust tracking, especially for wider fields of view or occluded objects. Multi-camera tracking is a correspondence problem between objects seen from different views at the same time or with a fixed time latency. It implies that all views of the same object should be given the same label.

Khan and Shah [12] presented a system based on the field of view lines of the camera to establish equivalence between views of the same object in different cameras. These lines were used to resolve any ambiguity between multiple tracks. Cai and Aggarwal [13] used only relative calibration between cameras to track objects over a long distance. Dockstader and Tekalp [14] introduced a distributed, real-time computing platform to improve feature-based tracking in the presence of articulation and occlusion. Their contribution was to perform both spatial and temporal data integration within a unified framework of 3D position tracking to provide increased robustness to temporal feature point occlusion. Lee [15] recovered the full 3D relative positions of the cameras and the domain plane of activity in the scene using only the tracks of moving objects.

1.3. Object classification

Despite the significant amount of literature on video surveillance, little work has been done on the task of classifying objects as single person, people group, or vehicle. Tan et al. [16] developed an algorithm to localize and recognize vehicles in traffic scenes. Lipton et al. [17] classified moving targets from a video stream into human, vehicle, or background regions. They used dispersedness value as a classification metric based on the assumption that humans were, in general, smaller than vehicles and had more complex shapes. This assumption became somewhat tenuous in cases where the humans were closer than the vehicles to the camera, where humans grouped together, or when vehicles were occluded. Foresti [18] described a visual surveillance system to classify moving objects into car, motorcycle, van, lorry, or person. In his system, object classification was based on a statistical morphology operator, the spectrum, which was used to index large image databases containing multiple views of different objects.

1.4. Overview of this paper

In this paper, we address the issues of tracking and classifying moving objects in an outdoor environment using the fusion of features and cameras. Our tracking objective is to establish a correspondence between the image structures of consecutive frames over time to form persistent object trajectories. Our tracking system integrates spatial position, shape, and color. This integration makes the tracker insensitive to background changes, motion interruption, and object orientation. To resolve the object occlusion, we fuse the trajectories from multiple cameras into a position and velocity in real world coordinates. This fusion is done by an extended Kalman filter, which enables our tracker to switch from one camera's observation to the other when the target object is occluded from view. Our underlying assumption for using the Kalman filter to fuse data is that there is a mathematical relationship between the target object's image positions in two synchronized cameras [19]. Furthermore, measurements from two synchronized cameras provide enough information to estimate the state variables of the system, the position and velocity in real world coordinates.

After obtaining an accurate description of the observed object, we classify the objects and update the templates, taking into account any occlusion. Our paper presents two robust classification metrics to classify the target object into single person, people group, or vehicle, namely the variance of motion direction and the variance of compactness. These two metrics are independent of the target object size and orientation and the camera used. The classification allows our tracker to know what kinds of objects are moving in the scene and to detect when two people or more come together to form a group, or separate from each other, dissolving a group.

The remainder of this paper is organized as follows. Section 2 describes tracking moving objects using a single camera. Classification metrics are derived in Section 3. Section 4 provides experimental results to demonstrate the accuracy and discusses the problems of tracking using a single camera. The associated classification results are also presented in this section. Section 5 applies an EKF to take advantage of multiple cameras. Experimental results using EKF are given in Section 6. Finally, Section 7 presents conclusions.

2. Single camera tracking

The first step in tracking objects is to segment the objects from the background. We use background subtraction at the expense of updating the background [20,21]. A pixel-wise median filter with L frame length is employed to build the background under the assumption that a moving object would not stay at the same position for more than $0.5L$ frames. L is typically of the order of 20. If the object were to

stay still more than $0.5L$ frames, it would be incorporated into the background. A median filter can build the background even when moving objects exist in the scene, but usually requires a large amount of memory to save L frames at a time. So, it is applied only when we detect a large new blob that lasts for η_b frames, $\eta_b = 5$ in our examples.

Background subtraction is performed in color and in edge density. We subtract the foreground from the background in each RGB color channel and then take the maximum absolute values of these three differences as the difference value in color space.

We do a similar subtraction using edge density values, instead of color values, and thus obtain the difference value in edge density. The edge density is defined as the average edge magnitude in a window.

The binary foreground pixels are the jointed pixels after the above two subtractions. We segment the foreground into several isolated blobs by an eight-connected algorithm. We assume that each initial blob has a moving object, which may be a person, a vehicle, or a people group.

2.1. Feature extraction

We extract four types of features for each moving blob. These features are used in object tracking except for the variance of motion direction, which is used for classification purposes.

The centroid of a blob $(\bar{M}, \bar{N})$ tells us the spatial position of the blob. It is the average position of all pixels in the blob. Since an object will not move far from its last position from one frame to the next, its centroid provides us with a strong and useful feature for tracking objects.

Shape features provide us with shape-based information about the objects. An object's shape generally does not change much between consecutive frames. We select four features—length and width, area, compactness, and orientation. The object orientation is defined as the angle between the major axis and the horizontal axis. An object with a more complex shape is more likely to change its compactness than an object with a simple shape, assuming that the same segmentation method is used. We define these features by:

Length and width (L, W) and area A

$$\text{Compactness} \quad C = \frac{4\pi \times \text{Area}}{\text{Perimeter}^2} \tag{1}$$

where Perimeter is the number of all boundary pixels including the boundaries of inside holes.

$$\text{Orientation} \quad \theta = \frac{1}{2}\arctan\left(\frac{2u_{11}}{u_{20} - u_{02}}\right) \tag{2}$$

where $u_{pq} = \sum_m \sum_n (m-\bar{M})^p (n-\bar{N})^q$ for $p, q = 0,1,2$.

The use of color as a feature allows us to track objects when shape information is not reliable. Color is independent of the object size and orientation and especially useful when we can detect only partial objects. However, color is likely to change with the lighting. Most previous research [22–24] has concentrated on how to stabilize color in some color space, e.g. HSV, normalized RGB, and YIQ. In our work, we use principal component analysis (PCA) to extract the color feature of the object. PCA is also called eigenspace decomposition and is applied to reduce the number of dimensions of the working space. It projects d-dimensional data onto a lower-dimensional subspace in a way that is optimal under a least square error criterion. We choose the axis that maximizes the variation of the data as our principal axis. This axis is the eigenvector corresponding to the maximum eigenvalue. We define the color similarity measure between objects and templates to be the transformation between their principal axes. All pixel RGB color values are collected from each blob. Three color channels of a pixel are denoted by a 1×3 vector $rgb = [R\ G\ B]$. The average color of all η pixels is given as

$$u_{rgb} = \frac{1}{\eta}\sum_{i=1}^{\eta} rgb_i \tag{3}$$

Then, a new vector is defined as $\hat{rgb} = rgb - u_{rgb}$. By concatenating all η pixels, an $\eta\times3$ matrix RGB is formed to express the red, green, and blue components of the full data set as $RGB = [\hat{rgb}_1; \hat{rgb}_2; \cdots \hat{rgb}_\eta]$. Next, a 3×3 covariance matrix Σ is calculated by

$$\Sigma = \frac{1}{\eta} RGB^t RGB \tag{4}$$

and the eigenvector and eigenvalue matrices are computed as $\Sigma\Phi = \Lambda\Phi$. We take the eigenvector Φ_1 corresponding to the largest eigenvalue Λ_1 as the principle axis. If two blobs have similar color, their principle axes should be similar.

After we extract the above features for each object blob, the feature vector $R_{i,K}$ will represent the blob, where $R_{i,k} = [\bar{M}, \bar{N}, L, W, A, C, \theta, \Phi_1]$. Our tracking is based on the extracted feature vector instead of the blob itself.

2.2. Object tracking

Our tracking process compares the feature vector $R_{i,k}$ with all templates $T_{i,k-1}$ $(i = 1,2\ldots\tau)$. If a match is found, then the template is updated for the next match through an adaptive filter of the form

$$T_{i,k} = (1-\beta)T_{i,k-1} + \beta R_{i,k} \tag{5}$$

where the learning parameter β depends upon how fast the template is updated. If no match is found for several successive frames, then a new candidate template $T_{\tau+1,k}$ will be created. A template will be deleted from the candidate template list if it is not matched with any object for successive L frames, the length of the median filer. A hierarchical matching process is performed in the order of

centroid, shape, and color. As soon as a match occurs, the process of matching terminates for the given blob.

In general, an object's next position will be in the neighborhood of its current position. We calculate the distances from a target object to all templates and sort the distances from the minimum to the maximum, namely $d_{\min}$, d_2,…$d_{\max}$. Three thresholds, Th_1, Th_2, and Th_3, are used in our centroid matching. An object is likely to be a new object if its $d_{\min} > \mathrm{Th}_1$. A match occurs if a distance is the $d_{\min}$ such that $d_{\min} < \mathrm{Th}_2$. Considering the case when occlusion occurs, we add a third constraint to avoid a possible mismatching, the second minimum distance $d_2 > d_{\min} + \mathrm{Th}_3$. This constraint prevents mismatching if there are two or more objects at a distance less than threshold Th_2 from the template. If the centroid matching procedure fails to match the target object to any template, then the next matching procedure, shape, or color matching, is necessary. To reduce the possibility of a false match, we compare the target object's shape or color only with templates that are the first four minimum distances from the object.

We compute the distance from the shape feature vector $[A, C, \theta]$ to the template as

$$\mathrm{Dis} = (A/A_{\mathrm{temp}} - 1)^2 + (C/C_{\mathrm{temp}} - 1)^2 + (\theta/\theta_{\mathrm{temp}} - 1)^2 \tag{6}$$

where A_{temp}, C_{temp}, and θ_{temp} are the area, compactness, and orientation of the template, respectively. The object is assigned to the template that yields the least distance if the distance is less than a predetermined threshold. To achieve a good measure of dissimilarity using distance, we normalize the shape feature prior to calculating the distance.

We use a similar function to compare the angles between the principle axes of the color of the object and the template. The normalized inner product

$$S(\Phi_{\mathrm{R}}, \Phi_{\mathrm{T}}) = \frac{\Phi_{\mathrm{R}}^t \Phi_{\mathrm{T}}}{||\Phi_{\mathrm{R}}||||\Phi_{\mathrm{T}}||} \tag{7}$$

is used as the similarity function. The value of this metric is larger when blob Φ_{R} and template Φ_{T} are similar. This measurement, which is the cosine of the angle between Φ_{R} and Φ_{T}, is invariant under rotation and dilation although it is variant under translation and general linear transformation. If Φ_{R} and Φ_{T} have a small angle between them, they form a match.

If an object goes through the above three procedures and still does not match any templates, a new candidate template for this object is generated. This candidate will turn into a true template after lasting for several successive frames. During this interim period, the temporary candidate template cannot be used for matching, and any matching of the object to another template will eliminate the candidate template. If two or more objects match the same template, a decision is made as to which one will be used to update the template. Usually, we use the object with the minimum centroid distance from the template.

3. Object classification

The motion feature is an efficient way to track humans and vehicles [25,26]. Vehicles are more consistent in their motion because they are rigid objects, whereas humans shift some parts of their bodies backwards when they move forward to maintain balance. So the variance of motion direction is employed to measure motion consistency. We derive this feature from optical flow.

Optical flow is widely used to estimate motion based on the assumption that the change in image brightness in the sequence is due only to motion. This assumption leads to a brightness constraint equation as

$$\frac{\mathrm{d}I}{\mathrm{d}t}(x,y,t) + \frac{\partial I}{\partial x}(x,y,t)\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}(x,y,t)\frac{\mathrm{d}y}{\mathrm{d}t} = 0 \tag{8}$$

where $I(x, y, t)$ is the image intensity as a function of space (x, y) and time t. This equation does not determine the flow field since it only has one restriction for two parameters dx/dt and dy/dt. In order to obtain a unique solution, an additional smoothness constraint is imposed as

$$\arg\min\left\{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2\right\} \tag{9}$$

By Lagrange optimization, the solutions are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \nu_{x,t} = \frac{\frac{\mathrm{d}I}{\mathrm{d}t}\frac{\partial I}{\partial x}}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{10}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \nu_{y,t} = \frac{\frac{\mathrm{d}I}{\mathrm{d}t}\frac{\partial I}{\partial y}}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{11}$$

where $\nu_{x,t}$, and $\nu_{y,t}$ are the velocities along the x-axis and y-axis at time t, respectively. The weighted velocity sums of three successive frames with weights, a_{-1}, a_0, and a_1 are

$$V_{x,t} = a_{-1}\nu_{x,t-1} + a_0\nu_{x,t} + a_1\nu_{x,t+1} \tag{12}$$

$$V_{y,t} = a_{-1}\nu_{y,t-1} + a_0\nu_{y,t} + a_1\nu_{y,t+1} \tag{13}$$

Then, a Gaussian filter G is applied, yielding

$$V_{x,t}^* = V_{x,t} \otimes G \quad \text{and} \quad V_{y,t}^* = V_{y,t} \otimes G \tag{14}$$

where the symbol $\otimes$ is a convolution operator.

We define motion direction as

$$\delta = \arctan \quad 2(V_{y,t}^*/V_{x,t}^*) \tag{15}$$

where arctan 2 is the four quadrant arctangent function. We calculate δ for each pixel in an object blob and compute the variance σ_δ^2 of all δs in the blob. Our experiments show that σ_δ^2 is an efficient metric to distinguish a single person.

The variance of motion direction σ_δ^2 cannot discriminate a people group from a vehicle. We added the variance of compactness σ_c^2 into consideration based on the observation that the shape of a people group tends to change

dramatically, which is measured by the variance of compactness over frames, denoted as σ_c^2. The σ_c^2 is calculated using the η_c compactness values of a tracked object over η_c frames. $\eta_c = 20$ is used in our experiments. The tracked object has a compactness value in each frame where it has been tracked.

A simple yet efficient classifier using two classification metrics, σ_δ^2 and σ_c^2, is designed to classify the moving objects into three categories: single person, people group, and vehicle. Two thresholds $\mathrm{Th_s}$ and $\mathrm{Th_g}$ are estimated from the training data. A single person is first categorized by its large variance of motion direction as $\sigma_\delta^2 > \mathrm{Th_s}$. Then, a people group is differentiated from a vehicle by its large variance of compactness over frames as $\sigma_c^2 > \mathrm{Th_g}$. The remaining objects are classified as vehicles. In our classification, a bicycle is regarded as a vehicle.

4. Experiments in single camera tracking and classification

We used the PETS2001 datasets and other videos to evaluate the single camera tracking system. PETS2001 datasets were provided by the second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. Each video is digitized with a frame rate of 25 frames/s. Datasets include moving people and vehicles. These videos are challenging in terms of significant lighting variation, occlusion, and scene activity.

Figs. 1–3 show some sample frames from test videos. The van in Fig. 1(a) stayed still for a long time; hence, it was merged into the background. It suddenly started in the next frame. A 20-frame median filter updated the background, and a new candidate template was created, shown in Fig. 1(b) by a rectangular blob. In Fig. 1(c), this new candidate lasted more than three successive frames and thus began to be tracked as a moving object.

In Fig. 2(a), a car, which was automatically classified as vehicle, approached a tree. In Fig. 2(b), the car appeared to be separated into two parts by the tree. The system recognized that these parts were the same. Fig. 2(c) shows the car emerging from behind the tree. We detected occlusion by comparing the blob size. If the size reduction exceeded a certain threshold for three successive frames, an occlusion was deemed to have occurred. In this case, only the centroid components of the template were updated. This was done using a linear prediction by assuming the velocity same as the preceding frame.

Fig. 3 presents an example of how to track several objects which group together and then come apart. In Fig. 3(a), a bicycle was passing a car. The bicycle blended with the car in Fig. 3(b). This blending ended at Fig. 3(c), and the objects again were tracked individually.

A classification example is drawn from the PETS2001 sequence with a tree presented earlier in Fig. 2. The variance of motion direction σ_δ^2 and the variance of compactness σ_c^2 are used as the classification metrics. Fig. 4(a) and (b) shows the variance of motion direction and the compactness of a single person, a people group, a vehicl,e and a bicycle. A single person is characterized by a high σ_δ^2 while a people group has a high σ_c^2. The variance of motion direction of a single person is above 1.2 on average, as shown in Fig. 4(a), while those of others are lower than 0.9. A threshold of variance of motion direction, $\mathrm{Th}_s = 1.0$, can be chosen so that a single person is detected when $\sigma_\delta^2 > 1.0$. It is clear from Fig. 4(b) that the variance of compactness of a people group is as high as 0.18, while those of other classes are lower than 0.02. Therefore, a threshold, $\mathrm{Th_g} = 0.1$ can classify a people group by $\sigma_c^2 > 0.1$. A bicycle is regarded as a vehicle in the classification. The classification error is less than 5% in our experiments [27].

Figs. 5–7 show several examples where individual features failed to track objects. However, by integrating the centroid, shape and color features, we can track the objects accurately. Fig. 5 shows a sudden action. The predicted position based on constant velocity failed to track the object since it quickly changed the velocity. Fig. 6 shows a large group of persons. The constant position predication did not work because many objects changed their positions when several persons met together. Fig. 7 shows a dynamic background with shadow. The shape feature was not reliable because the shadow was included in

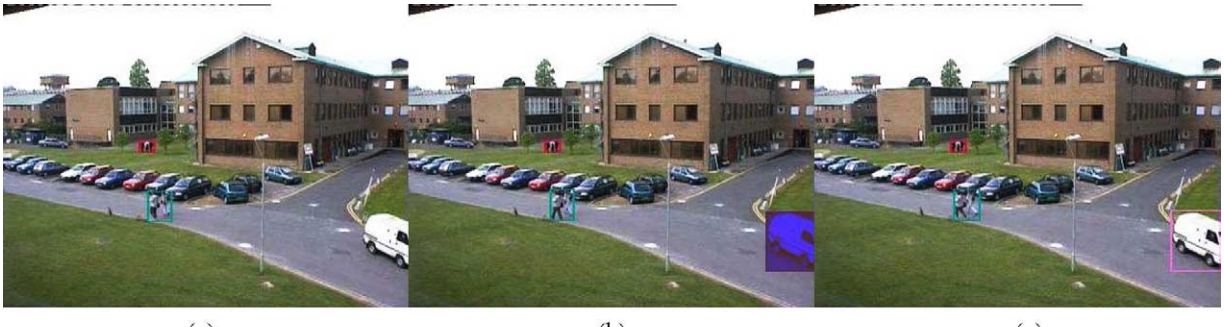


Fig. 1. (a) The van is still. (b) When the van moves, a new candidate template is created. (c) The candidate template lasts for three frames and thus becomes a true template and tracking begins.

(a) (b) (c)

Fig. 2. (a) A car approaches the tree. (b) The car appears to be in two parts. (c) The car emerges from behind the tree.

(a) (b) (c)

Fig. 3. (a) Before grouping. (b) Grouping. (c) After grouping.

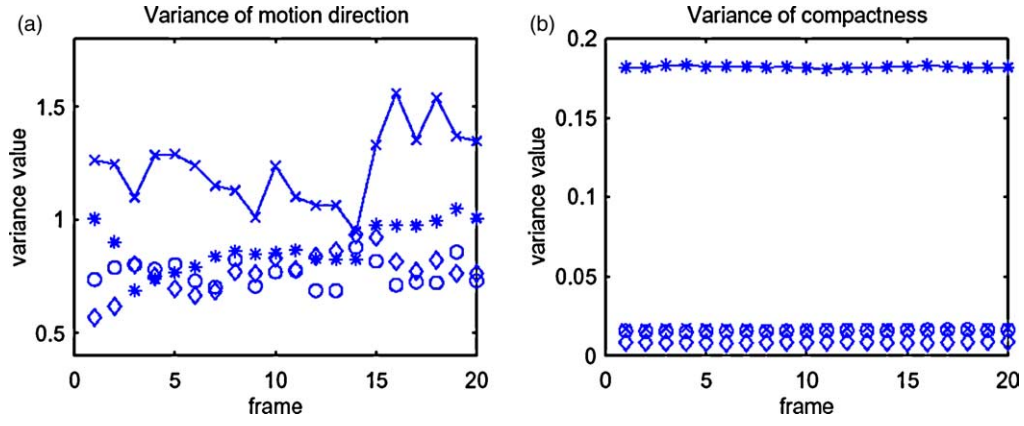
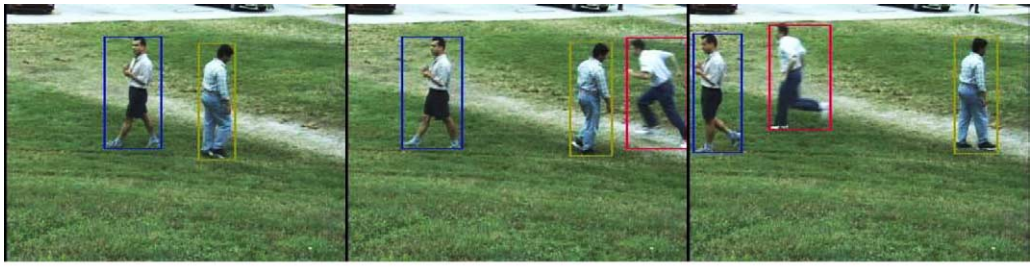


Fig. 4. ◇, vehicle; ○, bicycle; ×, single person; *, people group.



(a) (b) (c)

Fig. 5. Sudden action.

(a) (b) (c)

Fig. 6. Large people group.

(a) (b) (c)

Fig. 7. Dynamic background.

the shape feature. Our integration of multiple features makes the tracker more robust in these three sequences. All these sequences were tracked correctly as shown in Figs. 5–7. Classification results of Fig. 7 are shown in Fig. 8. Given the thresholds $\mathrm{Th_s}=1.0$ and $\mathrm{Th_g}=0.1$, the single person is detected, as its $\sigma_\delta^2 = 1.4 > \mathrm{Th_s}$, and the people group with three people is also detected, as its variance of compactness is 0.14, greater than the chosen threshold $\mathrm{Th_g}$.

In summary, the proposed system was used to analyze several videos and gave promising results. The background updating met problems when the lighting changed gradually. In this situation, our median filter kept updating the background without holding a static background. Vehicles were tracked successfully in part because of their large size. People tracking worked reliably through all sequences as long as the people were sufficiently isolated from each other. Groups of people were not handled well in that they significantly occluded each other's outlines in the image. The overall performance of single camera tracking was good. We accurately tracked 13 objects out of 17 moving objects from the PETS 2001's videos, with an accuracy of 76%. Among the failed objects, three were due to occlusion. Our own testing videos gave an accuracy of 82% from 50 objects.

5. Multiple camera tracking

In the previous sections, we developed a framework to track objects using a single fixed camera. Although we found the results of our single-camera tracker to be encouraging, there are some unresolved problems, mainly

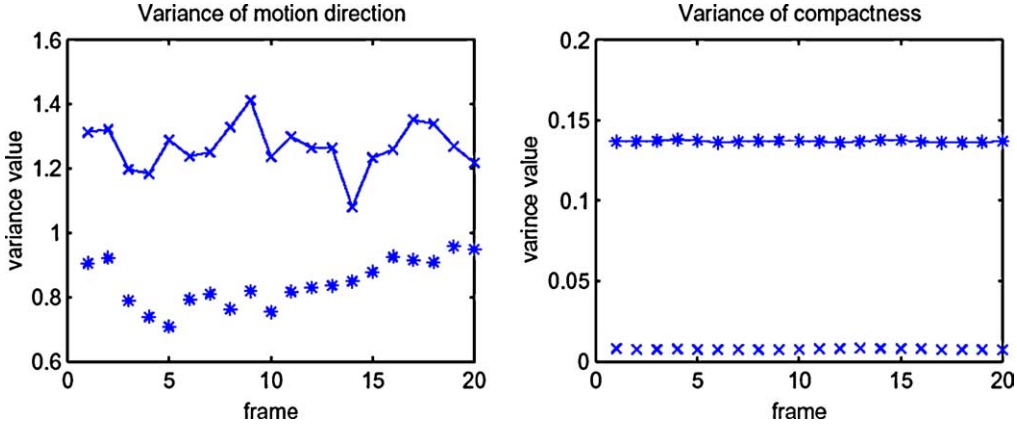


Fig. 8. Classification results from Fig. 9; ×, single person; *, people group.

due to object occlusion. For example, the bicycle shown in Fig. 2 was tracked as a different object once it passed behind a tree.

This section focuses on developing a methodology for tracking objects in the views of two fixed cameras. We consider the tracking problem as a dynamic target tracked by two disparate cameras, each with different measurement dynamics and noise characteristics. We combine the multiple camera views to obtain a joint tracking that is better than the single camera tracking in handling occlusion. In this paper, we fuse the individual camera observations to obtain combined measurements and then use a Kalman filter to obtain a final state estimate based on the fused measurements [28]. Measurement fusion is accomplished by increasing the dimension of the observation vector of the Kalman filter, based on the assumption that there is a mathematical relationship between the positions of an object Q in two disparate camera views. Since the mapping from one camera's view to the other view is non-linear, an extended Kalman filter is used to estimate the state vector [29].

5.1. Camera calibration

In order to determine the mathematical relationship between the observations from two different views, we calibrate the two cameras using five coplanar control points [30]. Another calibration method using three coplanar points has been reported to track the ground point [31]; however, since the ground point is not always reliable due to noise, we use five points for our calibration. Fig. 9 shows the five coplanar control points. Points A, B, D, and E are four vertices of a rectangle and point C is the center of the rectangle. For our application, an accurate camera calibration simply means that the 2D image coordinate can be properly predicted given the 3D location of the object.

The transformation from the real world position (x_w, y_w, z_w) to the camera 3D coordinates (x_c, y_c, z_c) is given by

Fig. 9. The control points on the ground plane ($z=0$).

a rotation matrix and a translation vector as

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} T_{11} & T_{21} & T_{31} \\ T_{12} & T_{22} & T_{32} \\ T_{13} & T_{23} & T_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} T_{41} \\ T_{42} \\ T_{43} \end{bmatrix} \tag{16}$$

where the rotation matrix and the translation vector are called the homogeneous transformation T. The transformation from 3D camera coordinates (x_c, y_c, z_c) to the ideal (undistorted) image coordinate (Q_x, Q_y) is obtained using perspective projection with pinhole camera geometry as

$$Q_x = \frac{x_c}{z_c} f \quad \text{and} \quad Q_y = \frac{y_c}{z_c} f \tag{17}$$

where f is the effective focal length.

Using the calibration technique presented by [30], we know the homogeneous transformation T, focal length f_T and the image center (T_x, T_y) for camera I and for camera II, S, f_s and (S_x, S_y). Therefore, we have the following equations for image position (Q_{1x}, Q_{1y}), (Q_{2x}, Q_{2y}) and the real world position (x_w, y_w, z_w)

$$\begin{aligned} Q_{1x} &= \frac{x_w T_{11} + y_w T_{21} + z_w T_{31} + T_{41}}{x_w T_{13} + y_w T_{23} + z_w T_{33} + T_{43}} f_T + T_x \\ &= \frac{x_{c,1}}{z_{c,1}} f_T + T_x \\ Q_{1y} &= \frac{x_w T_{12} + y_w T_{22} + z_w T_{32} + T_{42}}{x_w T_{13} + y_w T_{23} + z_w T_{33} + T_{43}} f_T + T_y \\ &= \frac{y_{c,1}}{z_{c,1}} f_T + T_y \\ Q_{2x} &= \frac{x_w S_{11} + y_w S_{21} + z_w S_{31} + S_{41}}{x_w S_{13} + y_w S_{23} + z_w S_{33} + T_{43}} f_S + S_x \\ &= \frac{x_{c,2}}{z_{c,2}} f_S + S_x \\ Q_{1y} &= \frac{x_w S_{12} + y_w S_{22} + z_w S_{32} + S_{42}}{x_w S_{13} + y_w S_{23} + z_w S_{33} + S_{43}} f_S + S_y \\ &= \frac{y_{c,2}}{z_{c,2}} f_S + S_y \end{aligned} \tag{18}$$

Knowing the calibration of cameras, we are able to merge the object's tracking from two different views into a real world coordinate view. The 3D tracking data is the state vector X_k, which cannot be measured directly. The object's positions in the two camera views are the measurement. We assume a constant velocity between two consecutive frames.

The dynamic equation related to the state vector X_k is described as follows

$$X_{k+1} = \Phi_k X_k + W_k \tag{19}$$

where $X_k = [x_w y_w z_w \dot{x}_w \dot{y}_w \dot{z}_w]^T$, the spatial position and velocity in real world coordinates, ΔT denotes the time

step in the transition matrix

$$\Phi_k = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

and W_k is white noise with a known covariance matrix. The subscript k in the notation denotes the frame number. It is often omitted for a general frame.

We derive the observation equation from the object's positions in two camera views. Let $[Q_{1x,k},Q_{1y,k}]$ and $[Q_{2x,k}, Q_{2y,k}]$ be the image positions of object Q in two camera views at kth frame. The observation vector

$$Z_k = [Q_{1x,k}, Q_{1y,k}, Q_{2x,k}]^{\mathrm{T}} = h_k(X_k) + V_k \tag{21}$$

where V_k is white noise with known covariance matrix, and h_k is a non-linear function relating the state vector X_k to the measurement Z_k.

The dynamic equation is linear but the measurement equation is non-linear, so the EKF is used to estimate the state vector X_k. Expanding in a Taylor series and neglecting higher order terms

$$\begin{aligned} Z_k &= h_k(X_k) + V_k \\ &= h_k(\hat{X}_{k|k-1}) + H'_k(X_k - \hat{X}_{k|k-1}) + V_k \\ &= H'_k X_k + V_k + h_k(\hat{X}_{k|k-1}) - H'_k \hat{X}_{k|k-1} \end{aligned} \tag{22}$$

where $\hat{X}_{k|k-1}$ is the estimation of the state vector from $\hat{X}_{k-1}$ and H'_k is the Jacobian matrix of partial derivations of $h_k(X_k)$ with respect to X_k.

The EKF recursive equations are

Update of the state estimation

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + L_k[Z_k - h_k(\hat{X}_{k|k-1})] \tag{23}$$

Prediction of states

$$X_{k+1|k} = \Phi_k X_{k|k} \tag{24}$$

Kalman gain matrix

$$L_k = \Sigma_{k|k-1} H_k (H'_k \Sigma_{k|k-1} H_k + Q_k)^{-1} \tag{25}$$

Update of the covariance matrix of states

$$\Sigma_{k|k} = \Sigma_{k|k-1} - \Sigma_{k|k-1} H_k (H'_k \Sigma_{k|k-1} H_k + Q_k)^{-1} H'_k \Sigma_{k|k-1} \tag{26}$$

Predication of the covariance matrix of states

$$\Sigma_{k+1|k} = \Phi_k \Sigma_{k|k} \Phi'_k + R_k \tag{27}$$

Initialization is provided by

$$\Sigma_{0|-1} = E[(X_0 - \bar{X}_0)(X_0 - \bar{X}_0)'] \quad \text{and} \quad \hat{X}_{0|-1} = \bar{X}_0.$$

The EKF uses a linearization of the state equations and the observation equations about the current best estimate of the state to produce minimum mean-square estimates of the state when W_k and V_k are white noise. When a filter is actually working, so that quantities trace $\Sigma_{k|k}$ and trace $\Sigma_{k|k-1}$ become available, one can use these as guides to $||X_k - \hat{X}_{k|k}||^2$ and $||X_k - \hat{X}_{k|k-1}||^2$, and this in turn allows estimating the amount of approximation involved.

5.2. Tracking initialization

Tracking initialization is a process of labeling a new object in the view of a camera. If this new object has a correspondence in the other camera, both objects should be assigned the same tag number. The reason behind tracking initialization is that when we recover the 3D coordinates of an object and project the estimated $(x_{\mathrm{w}}, y_{\mathrm{w}}, z_{\mathrm{w}})$ into Q_{2y}, the estimated Q_{2y} should be near the observed Q_{2y}. We initialized tracking by solving a constrained linear least squares problem, which is described as follows

$$\min_{[x_{\mathrm{w}}, y_{\mathrm{w}}, z_{\mathrm{w}}]} \left\| \begin{matrix} f_T x_{\mathrm{c},1} - Q_{1x} z_{\mathrm{c},1} \\ f_T y_{\mathrm{c},1} - Q_{1y} z_{\mathrm{c},1} \\ f_S x_{\mathrm{c},2} - Q_{2x} z_{\mathrm{c},2} \\ f_S y_{\mathrm{c},2} - Q_{2y} z_{\mathrm{c},2} \end{matrix} \right\|_2 \quad \text{such that } z > z_0 \tag{28}$$

where all the symbols are defined in Eq. (18). The sum of errors is minimum when all the objects in the view of one camera correctly find their correspondences in the other camera. Rearranging the above equations yields

$$\min_Y ||GY - D||_2^2 \quad \text{such that } z > z_0 \tag{29}$$

where

$$G = \begin{bmatrix} T_{11} f_T - Q_{1x} T_{13} & T_{21} f_T - Q_{1x} T_{23} & T_{31} f_T - Q_{1x} T_{33} \\ T_{12} f_T - Q_{1y} T_{13} & T_{22} f_T - Q_{1y} T_{23} & T_{32} f_T - Q_{1y} T_{33} \\ S_{11} f_S - Q_{2x} S_{13} & S_{21} f_S - Q_{2x} S_{23} & S_{31} f_S - Q_{2x} S_{33} \\ S_{12} f_S - Q_{2y} S_{13} & S_{22} f_S - Q_{2y} S_{23} & S_{32} f_S - Q_{2y} S_{33} \end{bmatrix},$$

$$Y = \begin{bmatrix} x_{\mathrm{w}} & y_{\mathrm{w}} & z_{\mathrm{w}} \end{bmatrix}^{\mathrm{T}}$$

and

$$D = \begin{bmatrix} Q_{1x} T_{43} - T_{41} f_T \\ Q_{1y} T_{43} - T_{42} f_T \\ Q_{2x} S_{43} - S_{41} f_S \\ Q_{2y} S_{43} - S_{42} f_S \end{bmatrix}.$$

We add a constraint $z > z_0$ since the moving object in the real world always has a non-zero height. This helps us to remove the shadow on the ground plane, whose height is zero. We regard the object Q_1 in camera I and the object Q_2 in camera II as the same object Q if the sum from the above

optimization problem is the minimum of all possible combinations.

6. Experiments in multiple camera tracking

Figs. 10 and 11 show two examples of multiple camera tracking. The dots are the observations in each camera and the lines are estimations from the EKF. x and y in the plots are horizontal and vertical coordinates. In Fig. 10, the projections from the real world trajectory estimated by EKF accurately fit the observations from the two cameras. The person appears in both views consistently, therefore, the trajectory is continuous. In Fig. 11, the car is occluded in the left view so its trajectory is interrupted for a while. However, the associated trajectory in the right view is continuous during that period, which can be used to predict the interrupted positions. The overall accuracy is 94% for the 17 moving objects in the PETS 2001's videos. Three objects lost in single camera tracking were tracked properly using two cameras. Only one object was lost for a short time in the multiple camera tracking.

There is some divergence between the projections and the observations in Fig. 11. One source of the divergence comes from the camera calibration and would be reduced by using more control points to calibrate the camera [30]. The other source is the time lag from the extended Kalman filter.

Our experiments demonstrate that the overall performance of multiple camera tracking is better than that of a single camera, especially when occlusion occurs. The EKF works well with most temporary occlusions, and the tracking initialization process can deal with long-term occlusion of more than 100 frames. Using multiple cameras, we can track the bicyclist in dataset II, which is occluded for a long time by the tree in camera I and thus would be difficult to track using only camera I. More accurate camera calibration would reduce the tracking error at the cost of having a more complex observation equation in the EKF. Multiple camera tracking relies on the objects existing in at least two camera views most of the time. When the target object is out of the field of one camera permanently, the tracking reduced to the single camera method.

7. Conclusions

In this paper, we have presented a system for tracking and classifying moving objects using single and multiple

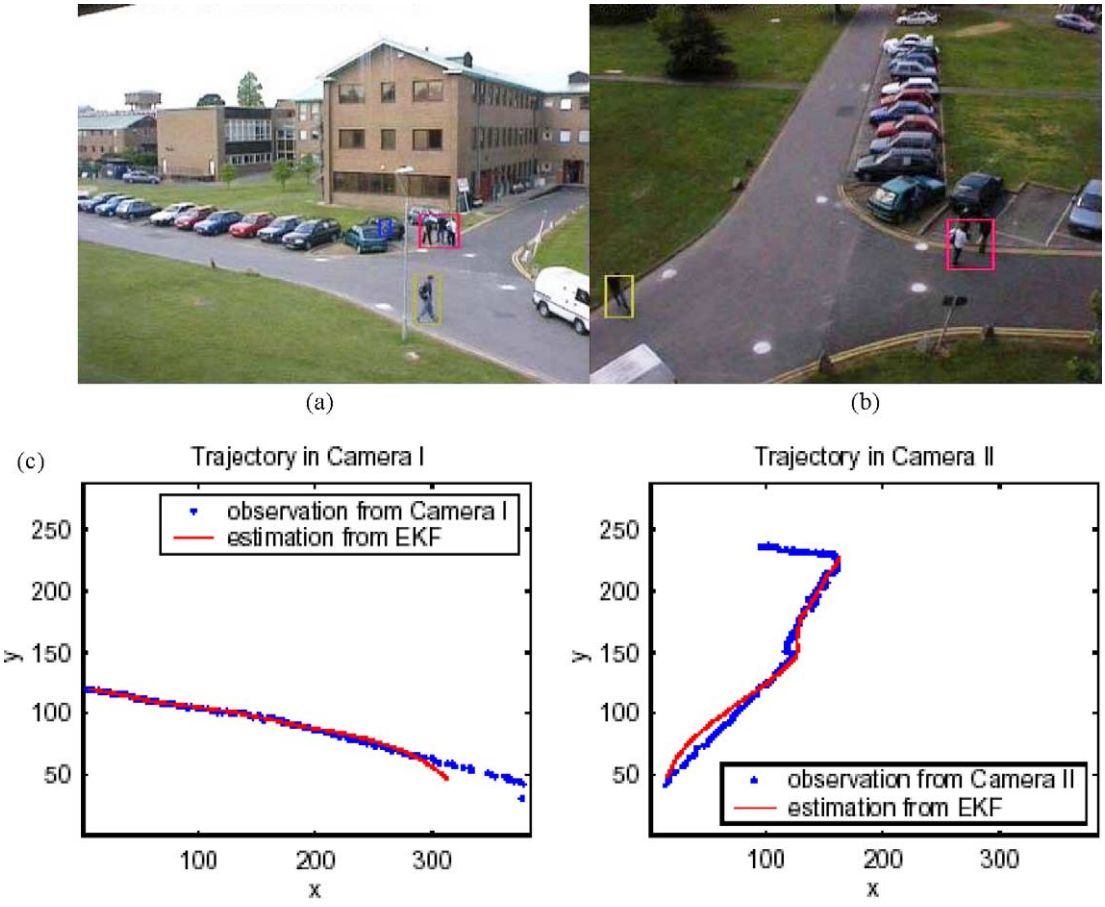


Fig. 10. Trajectory of the single pedestrian.

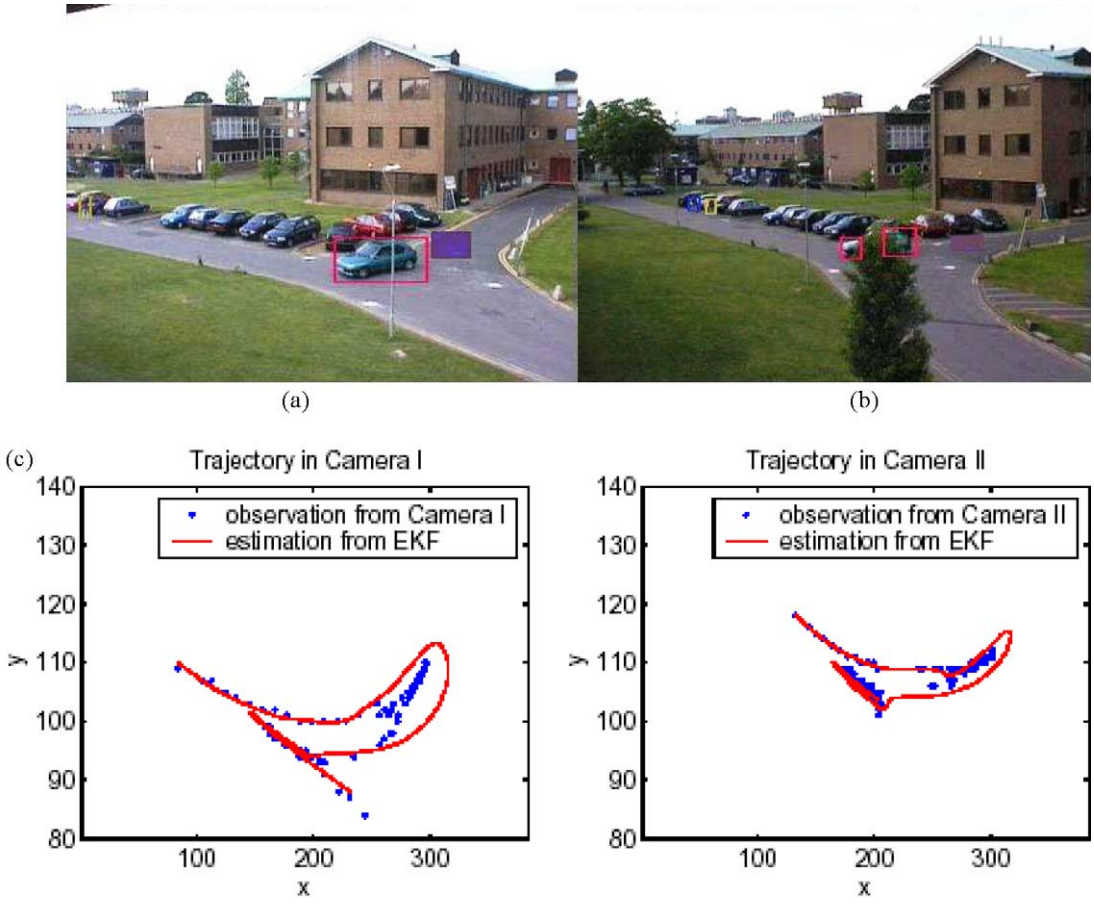


Fig. 11. Trajectory of the car with occlusion.

cameras in an outdoor environment. We combine spatial position, shape and color to achieve good performance in tracking people and vehicles. Principle component analysis is used to extract a color vector that is less sensitive to lighting changes. The variance of motion direction is robust for discriminating a single person. The variance of compactness can efficiently detect a people group. It is encouraging and useful that simple features are successful in tracking and recognition of moving objects. People tracking is more challenging than vehicle tracking due to the smaller object size and higher possibility of occlusion.

The extended Kalman filter fuses data from multiple cameras and performs quite well despite occlusion. However, the effectiveness of EKF may be reduced when occlusion happens in both camera views. Occlusions only in one camera view are handled successfully. An overall accuracy of 94% using multiple cameras was obtained for the PETS 2001 datasets, better than using a single camera. In addition, none of the thresholds or other parameters were changed when switching from single camera tracking to multiple camera tracking.

Based on our success in tracking multiple objects across multiple video streams, the reported research may be extended to recognizing moving object activities from multiple perspectives. Such a system would automatically monitor the moving objects, including humans and vehicles, classify activities, and maintain a video database, recording the time and location of motion.

Acknowledgements

An earlier version of this paper appeared in Handbook of Pattern Recognition and Computer Vision, C. H. Chen and P. S. P. Wang (Eds.), 3rd Edition, World Scientific Publishing Co., Singapore, pp. 499–524, 2005 [32].

References

[1] I. Haritaoglu, D. Harwood, L.S. Davis, W/sup 4/: real-time surveillance of people and their activities, IEEE Transaction on Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–830.

[2] Q. Cai, J.K. Aggarwal, Tracking human motion in structured environments using a distributed-camera system, IEEE Transaction on Pattern Analysis and Machine Intelligence 21 (11) (1999) 1241–1247.

[3] C. Stauffer, W.E.L. Grimson, Learning patterns of activity using real-time tracking, IEEE Transaction on Pattern Analysis and Machine Intelligence 22 (8) (2000) 747–757.

[4] S. Park, J.K. Aggarwal, A hierarchical Bayesian network for event recognition of human actions and interactions, ACM Journal on Multimedia Systems 10 (2) (2004) 164–179.
[5] J. Shi, C. Tomasi, Good features to track Proceeding of the International Conference on Computer Vision and Pattern Recognition, Seattle, Washington (1994) pp. 593–600.
[6] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto, Making good features track better, Proceedings of the International Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA (1998) pp. 178–183.
[7] T. Kaneko, O. Hori, Feature selection for reliable tracking using template matching, Proceeding of the International Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin (2003) pp. 796–802.
[8] C. Rasmussen, G.D. Hager, Probabilistic data association method for tracking complex visual objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 560–576.
[9] Y. Wu, T.S. Huang, A co-approach to robust visual tracking, Proceedings of Interational Conference on Computer Vision, Vancouver, Canada (2001) pp. 26–33.
[10] J. Triesch, C. Malsburg, Self-organized integration of adaptive visual cues for face tracking, Proceeding of the International Conference on Automatic Face and Gesture Recognition, Grenoble, France (2000) pp. 102–107.
[11] G. Rigoll, S. Eickeler, S. Muller, Person tracking in real-world scenarios using statistical method, Proceeding of the International Conference on Automatic Face and Gesture Recognition, Grenoble, France (2000) pp. 342–347.
[12] S. Khan, M. Shan, Consistent labeling of tracking objects in multiple cameras with overlapping fields of view, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1355–1360.
[13] Q. Cai, J.K. Aggarwal, Tracking human motion using multiple cameras, Proceeding of the International Conference on Pattern Recognition, Vienna, Austria (1996) pp. 68–72.
[14] S.L. Dockstader, A.M. Tekalp, Multiple camera tracking of interacting and occluded human motion, Proceeding of the IEEE 89 (10) (2001) 1441–1455.
[15] L. Lee, R. Romano, G. Stein, Monitoring activities from multiple video streams: establishing a common coordinate frame, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 758–767.
[16] T.N. Tan, G.D. Sullivan, K.D. Baker, Efficient image gradient based object localization and recognition, Proceeding of the International Conference on Computer Vision and Pattern Recognition, San Francisco, California (1996) pp. 397–402.
[17] J. Lipton, H. Fujiyoshi, R.S. Patil, Moving target classification and tracking from real-time video, Proceeding of the IEEE Workshop on Application of Computer Vision, Princeton, NJ (1998) pp. 8–14.
[18] G.L. Foresti, A real-time system for video surveillance of unattended outdoor environments, IEEE Transactions on Circuits and System for Video Technology 8 (6) (1998) 697–704.
[19] K. Bradshaw, I. Reid, D. Murray, The active recovery of 3D motion trajectories and their use in prediction, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (3) (1997) 219–234.
[20] M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 1–15.
[21] D. Gutchess, M. Trajkovic, E. Cohen-Sola, D. Lyond, A.K. Jain, A background model initialization algorithm for video surveillance, Proceeding of the International Conference on Computer Vision, Vancouver, Canada, Vancouver, Canada, 2001 pp. 733–740.
[22] Q. Iqbal, J.K. Aggarwal, Retrieval by classification of image containing large manmade objects using perceptual grouping, Pattern Recognition 35 (7) (2002) 1463–1479.
[23] Y. Raja, S.J. Mckenna, S. Gong, Tracking and segmenting people in varying lighting conditions using colour, Proceeding of the International Conference on Automatic Face and Gesture Recognition, Nara, Japan (1998) pp. 228–233.
[24] C.R. Wren, A. Azarbayejani, T. Darrel, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[25] F. Bobick, J.W. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.
[26] K. Sato, J.K. Aggarwal, The detection and recognition of events in video, Computer Vision and Image Understand 96 (2) (2004) 100–128.
[27] Q. Zhou, J.K. Aggarwal, Tracking and classifying moving objects from video International Workshop on Performance Evaluation of Tracking and Surveillance, Kauai, Hawaii (2001).
[28] N. Strobel, S. Spors, R. Rabenstein, Joint audio–video object localization and tracking, IEEE Signal, Processing Magazine 18 (1) (2001) 22–31.
[29] B. Anderson, J. Moore, Optimal Filtering, Prentice-Hall, New Jersy, 1979.
[30] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE Journal of Robotics and Automation 3 (4) (1987) 323–344.
[31] Q. Zhou, J. Park, J.K. Aggarwal, Quaternion-based tracking of multiple objects in synchronized video, Proceedings of the International Symposium on Computer and Information Sciences, Antalys, Turkey (2003) pp. 430–438.
[32] Q. Zhou, J.K. Aggarwal, Tracking and classifying moving objects using single or multiple cameras in: C.H. Chen, P.S.P. Wang (Eds.), Handbook of Pattern Recognition and Computer Visionthird ed., World Scientific Publishing Co., Singapore, 2005, pp. 499–524.