

Method And System For Multi-Object Tracking

FIELD OF INVENTION

The present invention relates broadly to method and system for multi-object tracking in a video signal, in particular a video signal captured by a fixed camera, and to a data storage medium having stored thereon computer code means for instructing a computer system to execute a method for multi-object tracking in a video signal.

BACKGROUND

Real-time tracking objects of interest in image sequences is one of the challenging problems in video understanding. It is an essential part in many computer vision applications, such as video surveillance, media analysis, human computer interface, and video compression (e.g., object-based coding in MPEG-4). Many methods have been proposed for appearance-based visual object tracking in image sequences. Generally speaking, three components are included in the processing: *target representation*, *motion prediction*, and *object matching*. First, a model is used to characterize the distinctive appearance features of the target object. These appearance features of the target object should be consistent and discriminant from other objects through the image sequence. Existing models include blobs of homogeneous intensities or colors, feature points, contours, templates, color histograms and joint color-spatial distributions of the object regions.

Inter-frame motion of target objects is predicted by using dynamic models or motion models. The popular dynamic models are Kalman filter and particle filters. The motion of a rigid object can be estimated by using an explicit motion model such as geometric transformation, while motion of non-rigid objects can be computed with an implicit motion model, e.g., mean-shift. An observation model is used to evaluate the matching between the target and its candidate location in the coming frame. The new position of the target is determined as the mean location of the predicted candidates (*EAP: expected a posterior*) if a dynamic model is employed or the location of the maximum matching value (*MAP: maximum a posterior*) when a motion model is used.

The MAP methods are considered as *deterministic* tracking since a gradient descent algorithm, e.g., a mean-shift algorithm, is commonly used to find the maximum, and the EAP

methods are classified as *stochastic* tracking as randomly sampling in a time series state space is required. Roughly speaking, stochastic tracking is more robust than deterministic tracking since it is less likely to be trapped in local extrema, however, it is more computational expensive.

While tracking a single object in cluttered environments has been largely successful, tracking a varying number of non-rigid objects in crowds remains a challenging task. Among many others, there are three major difficulties for solving this problem. First, when objects merge into a group, the visual features for each object become ambiguous and uncertain. The distant objects can be occluded partially or even completely by the close objects. Second, the appearance of target objects may change drastically when they are in crowds due to the changes of poses and scales. For example, one standing person may sit down when he is partially occluded by another person. Third, the motion modes of target objects may change significantly in crowds, e.g., several separated persons may gather into a group, stay together for a while and then separate in different directions.

There has been an increasing number of proposals on multi-object tracking in this decade. The existing methods can roughly be classified into three categories: multiple instantiations of single trackers, multi-camera cooperation, and extended particle filters. A simple solution to the problem is to build multiple instantiations trackers for individual objects. Such methods perform well in the presence of simple occlusions with the help of Kalman filter and specified strategies to interpret the overlaps. In some proposals explicit segmentation of objects in a group using template matching is performed before tracking. Object models templates or color blobs of head, upper and lower parts of human body are learned for each isolated object. The segmentation of individuals in a group becomes more difficult when large variations of poses, scales, and motion patterns of objects are involved in the interaction.

Being aware of the difficulties caused by occlusion and overlapping, multi-camera systems are proposed. With a proper placement of multiple cameras, occlusion can be alleviated assuming at least one camera may capture a better view of each target object. The challenge for a multi-camera system is the calibration and the cooperation of multiple cameras to achieve a consistent tracking of each object since the views from different cameras can be very different.

Over the last few years, particle filters have been shown to be powerful tools for single object tracking. Attempts have also been made to extend particle filters to multi-object tracking. Hue *et al* extended the state space by concatenating the state vectors of fixed M objects. The likelihood measure is estimated under the assumption of conditional independence. The association for each component is assigned by a Gibbs sampler. Vermaak *et al* introduced a mixture of particle filters (MPF) where each component (object) is described by a cluster of

particle filters that form a part of the mixture. The filters in the mixture interact only through the computation of the importance weights. Okuma *et al* further developed a boosted particle filter (BPF) from MPF. In BPF, the proposal distribution for each object (hockey player) is estimated by integrating the object detection and the dynamic models.

To avoid the convergence of multiple modes of occluded objects into one cluster of the front object in the case of overlapping, MaccorMick and Blake proposed the exclusion principle in the likelihood estimation. In their work, the state is extended to a 2.1D model which contains a label of depth order. The likelihood measures for all the depth configurations between two overlapping objects are explicitly derived. A strategy of partition sampling is performed for all of the configurations. A similar approach has been used to estimate the likelihoods based on color histograms of two overlapping objects.

Considering the difficulties in deducing the occlusion relationship of multiple objects from images, introduction of 3D information about the background and target objects has been proposed. In one such proposal, a calibrated camera and a generalized-cylinder or 3-ellipsoid model of a standing human object are used, and the likelihood for a hypothesis configuration of multiple persons in a group is evaluated according to the 2D projections of their 3D positions on the ground plane. These methods successfully tracked multiple persons walking in crowds without large pose variations. However, the sufficient sampling of all the possible 3D configurations of multiple objects may lead to a significant increase of particle filters to obtain a proper distribution. When multiple objects gather together with various complex occlusions, the likelihoods of observing different objects are not independent. A few methods have sought to address such problem under a general mathematical framework and almost all of them are stochastic-based methods. A main disadvantage of these stochastic methods is their intensive computations which make them difficult for real-time video surveillance applications.

A need therefore exists to provide a method and system for multi-object tracking in a video signal that seek to address at least one of the above mentioned disadvantages.

SUMMARY

In accordance with a first aspect of the present invention there is provided a method of multi-object tracking in a video signal; the method comprising the steps of receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked;

generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and for each parent node having two or more child nodes,

- a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;
- b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;
- c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and
- iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;
- and then for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image; for each child node having two or more corresponding objects assigned thereto,
- d) sorting the corresponding objects according to estimated depth in said each child node in said second image;
- e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;
- f) updating the state and the visual content of the located corresponding object based on the second image;
- g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and
- iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

Step a) may comprise calculating visible portions of the respective corresponding objects in the first image to derive the estimated depths of the respective corresponding objects.

Step b) may comprise assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node based on a posterior probability evaluated by Bayes rule.

A posterior probability evaluated by Bayes rule may be based on principle colour representations (PCRs) of the corresponding object and said one child node respectively.

Step c) may comprise removing the visual content of the assigned corresponding object from the visual data associated with said one child node based on PCRs of the assigned corresponding object and said one child node respectively.

Step d) may comprise calculating visible portions of the respective corresponding objects in the second image to derive the estimated depths of the respective corresponding objects.

Step e) may comprise applying the means-shift calculation to locate the corresponding object having the lowest depth in said each child node based on gravity centres of pixels of each principle colour components in a PCR of the corresponding object in the second image.

Step g) may comprise removing the updated visual content of the located corresponding object from the visual data associated with said each child node based on PCRs of the located corresponding object and said each child node respectively.

The method may further comprise storing tracking data including the updated status and visual content of each corresponding object for a series of consecutive frames and detecting an event in the video signal based on the stored tracking data.

The method may further comprise the step of for each parent node having no child node, deleting the corresponding object.

The method may further comprise the step of for each parent node having only one child node, assigning all corresponding objects to said one child node.

The method may further comprise the step of for each child node having no corresponding object assigned thereto, check whether the object is disappeared, and if not, set a new corresponding object to said each child node.

The method may further comprise the step of for each child node having only one corresponding object assigned thereto, update the state and visual content of said one corresponding object from the visual data associated with said each child node.

In accordance with a second aspect of the present invention there is provided a multi-object tracking module for multi-object tracking in a video signal; the module comprising means for receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked; means for generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and the means for generating, for each parent node having two or more child nodes,

a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;

b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;

c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and

iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;

and then, for each child node having only one corresponding object assigned thereto, updating a state and the visual content of said one object based on the second image; for each child node having two or more corresponding objects assigned thereto,

d) sorting the corresponding objects according to estimated depth in said each child node in said second image;

e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;

f) updating the state and the visual content of the located corresponding object based on the second image;

g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and

iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

In accordance with a third aspect of the present invention there is provided a data storage medium having stored thereon computer code means for instructing a computer system to execute a method of multi-object tracking in a video signal; the method comprising the steps of receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked; generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and for each parent node having two or more child nodes,

a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;

b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;

c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and

iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;

and then for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image; for each child node having two or more corresponding objects assigned thereto,

d) sorting the corresponding objects according to estimated depth in said each child node in said second image;

e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;

f) updating the state and the visual content of the located corresponding object based on the second image;

g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and

iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

Also disclosed is a method of stationary object tracking in a video signal, the method comprising the steps of determining that a tracked moving object has become stationary over a sequence of frames; generating a template image of the stationary object based on at least one of the frames of the sequence; tracking a state of the stationary object based on a comparison of the template image with a current frame of the video signal; and switching to a moving object tracking algorithm using a same object ID if the state of the stationary object is determined as a start moving state.

The generating of the template image may be based on image data within a bounding box in the at least one of the frames of the sequence for the tracked object.

The tracking of the state may comprise the steps of determining a first difference measure between the template image and a corresponding region in the current frame; determining a second difference measure between respective corresponding regions in the current frame and a preceding frame; determining a visibility measure of the stationary object from the corresponding region in the current frame.

The tracking of the state further may comprise determining whether another tracked moving object overlaps the corresponding region in the current frame.

The tracking of the state further may comprise the steps of determining a motionless state if the first and second difference measures are below a first threshold value over a sequence of τ_p current frames; determining an occluded state if the first and second difference measures each exceed a second threshold value and the visibility measure falls below a third threshold value over the sequence of τ_p current frames, and another tracked moving object overlaps the corresponding region in the current frame; determining a removed state if the first and second difference measures each initially exceed the second threshold value and then the second difference measure falls below the first threshold value and the visibility measure falls below the third threshold value over the sequence of τ_p current frames, and no tracked moving object

overlaps the corresponding region in the current frame; determining an inner-motion state if the first and second difference measures each initially exceed the second threshold value and then the second difference measure then falls below the first threshold value and the visibility measure is above a fourth threshold value; and determining the start moving state if the first and second difference measures exceed and remain above the second threshold value and the visibility measure exceeds and remains above the fourth threshold measure over the sequence of τ_p current frames, and a spatial shift of the tracked stationary object.

The visibility measure may be determined based on principle colour representation.

The first and second difference measures may be determined based on a knowledge base of human perceived semantic meanings, an evaluation from real-world videos, or both.

Also disclosed is a system for object tracking in a video signal, the system comprising means for determining that a tracked moving object has become stationary over a sequence of frames; means for generating a template image of the stationary object based on at least one of the frames of the sequence; means for tracking a state of the stationary object based on a comparison of the template image with a current frame of the video signal; and means for switching to a moving object tracking algorithm using a same object ID if the state of the stationary object is determined as a start moving state.

Also disclosed is a data storage medium having stored thereon computer code means for instructing a computer system to execute a method of object tracking in a video signal, the method comprising the steps of determining that a tracked moving object has become stationary over a sequence of frames; generating a template image of the stationary object based on at least one of the frames of the sequence; tracking a state of the stationary object based on a comparison of the template image with a current frame of the video signal; and switching to a moving object tracking algorithm using a same object ID if the state of the stationary object is determined as a start moving state.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

Figure 1 shows a series of images illustrating adaptive background subtraction using the background updating method and system of the example embodiments.

Figure 2 shows a flow chart illustrating a method of context-based background updating for adaptive background subtraction in the example embodiment.

Figure 3 shows a series of images and histograms illustrating principle colour representation (PCR) in the example embodiment.

Figure 4 shows a schematic drawing illustrating directed acyclic graphs (DAGs) for regions in consecutive frames in the example embodiment.

Figure 5 shows a flow chart illustrating a method of multi-object tracking in a video signal in the example embodiment.

Figure 6 shows a flow chart illustrating a method of stationary object tracking in a video signal in the example embodiment.

Figure 7 shows a schematic drawing of an event detection system implementation using the example embodiment.

Figure 8 shows a graph illustrating a finite state machines (FSM) representation for event detection in the system implementation of Figure 7.

Figure 9 shows a schematic drawing of a computer system for implementing the example embodiment.

DETAILED DESCRIPTION

The described embodiment provides a novel 2½D method of multi-object tracking for real-time video surveillance. An appearance model, principal color representation (PCR), is applied to multi-object tracking. The PCR model characterizes the appearance of an object or a region with a few most significant colors. The likelihood of observing a tracked object in a foreground region is derived according to their PCRs. Based on the Bayesian estimation theory, multi-object tracking is formulated as a Maximum A Posterior (MAP) problem over all the tracked objects. With the foreground regions provided by background subtraction, the problem of multi-object tracking is decomposed into two subproblems: assignment and location.

By exploiting that the close and unoccluded objects have richer visual information than the distant or occluded ones, sequential solutions to the subproblems which process the objects in a group from the most visible to the least visible ones are derived according to the likelihoods estimated based on PCR. In the assignment step, each tracked object is assigned to a foreground region in the coming frame. When an object is assigned, its visual information will be excluded from the PCR of the region.

In the location step, multiple objects assigned to one region are located one-by-one according to their depth order. A two-phase mean-shift algorithm based on PCR is derived for locating objects. When an object is located, its visual information is excluded from the new position in the region. The operation of exclusion at the end of each iteration for assignment and location in the example embodiment can avoid multiple objects being trapped into the same region or position.

Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "calculating", "determining", "excluding", "generating", "assigning", "locating", or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be

appropriate. The structure of a conventional general purpose computer will appear from the description below.

In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention.

Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the preferred method.

The invention may also be implemented as hardware modules. More particular, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. Those skilled in the art will appreciate that the system can also be implemented as a combination of hardware and software modules.

The distinctive background objects (regions) in the example embodiment are classified into two categories:

Type-1 CBR: a facility for the public in the scene;

Type-2 CBR: a large homogenous region.

Contextual descriptors are developed to characterize the distinctive appearances of CBRs and evaluate the likelihoods of observing them. Different contextual background regions may have different appearance features. Some manifest significant structural features, while others may have homogeneous color distributions. The example embodiment employs *Orientation*

Histogram Representation (OHR) to describe the structural features of a region and *Principal Color Representation* (PCR) to describe the distribution of dominant colors. Let R_b^i be the i -th CBR in the empty scene $I(x)$, and $G(x)$ and $O(x)$ be the gradient and orientation images of $I(x)$, respectively. If the orientation values are quantized into 12 bins each covering 30° , the orientation histogram for R_b^i is defined as

$$H_b^i(k) = \sum_{x \in R_b^i} \mu_{T_O}(G(x)) \delta_k(O(x)) \quad (1a)$$

where $\mu_T()$ is a binary function on the threshold T and $\delta_k()$ is a delta function defined as

$$\delta_k(O(x)) = \begin{cases} 1, & \text{if } k \leq O(x)/30^\circ < k+1 \\ 0, & \text{otherwise} \end{cases} \quad (2a)$$

The OHR H_b^i is a simple and efficient variant of the robust local descriptor SIFT [1] for real-time processes. It is less sensitive to illumination changes and slight shift of object position.

By scanning the region R_b^i , a table of the PCR for the region can be obtained. The PCR for R_b^i is defined as

$$T_b^i = \{p_i, \{E_i^k = (c_i^k, p_i^k)\}_{k=1}^{N_i}\} \quad (3a)$$

where p_i is the size of R_b^i , c_i^k is the k -th most significant color of R_b^i and p_i^k is its significance value. The significance value is computed by

$$p_i^k = \sum_{x \in R_b^i} \delta(I(x), c_i^k) \quad (4a)$$

$\delta(c_1, c_2)$ is a delta function. It equals to 1 when the color distance $d(c_1, c_2)$ is smaller than a small threshold ε , otherwise, it is 0. The color distance used here is

$$d(c_1, c_2) = 1 - \frac{2 \langle c_1, c_2 \rangle}{\|c_1\|^2 + \|c_2\|^2} \quad (5a)$$

where $\langle \bullet, \bullet \rangle$ denotes the dot product [2, 3]. The principal color components E_i^k are sorted in descendent order according to their significance values p_i^k . The first N_i components which satisfy $\sum_{k=0}^{N_i} p_i^k \geq 0.95 p_i$ are used as the PCR of the region R_b^i , which means the principal colors in PCR cover more than 95% colors from R_b^i . PCR is thus efficient to describe large regions of distinctive colors.

A type-1 CBR in the example embodiment is associated with a facility which has a distinctive structure and colors in the image. Both OHR and PCR are used to characterize the type-1 CBR. Let $R_{b_l}^i$ be the i -th type-1 CBR in the scene. Its contextual descriptors are $H_{b_l}^i$ and $T_{b_l}^i$. A type-1 CBR has just two states: occluded (occupied) or not. The likelihood of observing a

type-1 CBR is evaluated on the whole region. Suppose the contextual descriptors of the region $R_i(\mathbf{x})$ from the corresponding position of R^i_{b1} in the current frame $I_t(\mathbf{x})$ are H_t and T_t . The likelihood of R^i_{b1} being exposed can be evaluated by matching $R_i(\mathbf{x})$ to R^i_{b1} . Based on OHR, the matching of $R_i(\mathbf{x})$ and R^i_{b1} is defined as

$$P_L(H_t|H_{b1}^i) = \frac{2 \langle H_{b1}^i, H_t \rangle}{\|H_{b1}^i\|^2 + \|H_t\|^2} \quad (6a)$$

If R^i_{b1} and $R_i(\mathbf{x})$ are similar, $P_L(H_t|H_{b1}^i)$ is close to 1, otherwise, it is close to 0.

Let $T_{b1}^i = \{p_{b1,i}, \{E_{b1,i}^k = (c_{b1,i}^k, p_{b1,i}^k)\}_{k=1}^{N_{b1}^i}\}$ be the PCR of R_{b1}^i and $T_t = \{p_t, \{E_t^n = (c_t^n, p_t^n)\}_{n=1}^{N_t}\}$ the PCR $R_t(\mathbf{x})$. According to the law of the total probability, the likelihood of $R_t(\mathbf{x})$ belonging to R_{b1}^i is

$$P(T_t|T_{b1}^i) = \sum_{k=1}^{N_{b1}^i} P(T_t|E_{b1,i}^k) P(E_{b1,i}^k|T_{b1}^i) \quad (7a)$$

The second term in the sum is the weight of the principal color $c_{b1,i}^k$ in the PCR of R^i_{b1} , i.e., $P(E_{b1,i}^k|T_{b1}^i) = p_{b1,i}^k/p_{b1,i}$. The first term is the likelihood based on the partition evidence of principal color $c_{b1,i}^k$. It is evaluated from the PCRs of R^i_{b1} and $R_t(\mathbf{x})$ as

$$P(T_t|E_{b1,i}^k) = \frac{1}{p_{b1,i}^k} \min \left\{ p_{b1,i}^k, \sum_{n=1}^{N_t} \delta(c_{b1,i}^k, c_t^n) p_t^n \right\} \quad (8a)$$

Then there is

$$P(T_t|T_{b1}^i) = \frac{1}{p_{b1,i}} \sum_{k=1}^{N_{b1}^i} \min \left\{ p_{b1,i}^k, \sum_{n=1}^{N_t} \delta(c_{b1,i}^k, c_t^n) p_t^n \right\} \quad (9a)$$

$P(T^i_{b1}|T_t)$ can be obtained in a similar way. Now the matching of R^i_{b1} and $R_t(\mathbf{x})$ based on PCR is defined as

$$P_L(T_t|T_{b1}^i) = \min\{P(T_t|T_{b1}^i), P(T_{b1}^i|T_t)\} \quad (10a)$$

Assuming that colors and the gradients are independent and different weights are used, the log likelihood of observing R^i_{b1} at time t is

$$L_{b1}^{i,t} = \omega_s \log P_L(H_t|H_{b1}^i) + (1-\omega_s) \log P_L(T_t|T_{b1}^i) \quad (11a)$$

where $\omega_s = 0.6$ is chosen empirically.

The type-2 CBRs in the example embodiment are large homogeneous regions. Only the PCR descriptor is used for each of them. Usually only part of a type-2 CBR is occluded when a foreground object overlaps it. The likelihood of observing a type-2 CBR is evaluated loc-

cally. Let $T_{b2}^i = \{p_{b2,i}, \{E_{b2,i}^k = (c_{b2,i}^k, p_{b2,i}^k)\}_{k=1}^{N_{b2}^i}\}$ be the PCR of the i -th type-2 CBR R_{b2}^i . Suppose $R_t(x)$ is a small neighborhood centered at x , e.g., a 5×5 window. The likelihood of $R_t(x)$ belonging to R_{b2}^i is defined as

$$P(R_t(x)|R_{b2}^i) = \frac{1}{|R_t(x)|} \sum_{s \in R_t(x)} B(I_t(s)|R_{b2}^i) \quad (12a)$$

where $|R_t(x)|$ is the size of the window and $B(I_t(s)|R_{b2}^i)$ is a Boolean function defined as

$$B(I_t(s)|R_{b2}^i) = \begin{cases} 1, & \exists c_{b2,i}^k, \delta(I_t(s), c_{b2,i}^k) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (13a)$$

The log likelihood that the pixel x in the current frame belongs to R_{b2}^i is

$$L_{b2}^{i,t}(x) = \log P(R_t(x)|R_{b2}^i) \quad (14a)$$

The *appearance model* of a type-1 CBR in the example embodiment consists of its OHR and PCR. For the i -th type-1 CBR R_{b1}^i , the *appearance model* is defined as $M_a(R_{b1}^i) = (H_{b1}^i, T_{b1}^i)$. The *spatial model* of R_{b1}^i is defined as its bounding box and the center point, i.e., $M_s(R_{b1}^i) = (B_{b1}^i, x_{b1}^i)$.

To adapt to lighting changes from day to night, besides the active appearance model $M_a(R_{b1}^i)$, a model base which contains up to K_b appearance models of R_{b1}^i is used. The models in the base are learned incrementally. The active appearance model is the one from the model base which best fits the current appearance of the CBR. The model base of

$$R_{b1}^i \text{ is } MB(R_{b1}^i) = \{M_a^k(R_{b1}^i)\}_{k=1}^{K_{b1}^i}, K_{b1}^i \leq K_b.$$

Natural lighting changes slowly and smoothly. Let D be a time duration of 3 to 5 minutes, not limiting, in the example embodiment (i.e. a long duration compared with the frame duration in the video signal). The times of observing the i -th type-1 CBR during the period are accumulated as

$$Z_{b1}^{i,c} = \sum_{t \in D} \mu_{T_{L1}}(L_{b1}^{i,t}) \quad (15a)$$

and the average of the likelihood values is

$$\bar{L}_{b1}^{i,c} = \frac{1}{Z_{b1}^{i,c}} \sum_{t \in D} L_{b1}^{i,t} \mu_{T_{L1}}(L_{b1}^{i,t}) \quad (16a)$$

where $L_{b1}^{i,t} > T_{L1}$ means R_{b1}^i is visible at time t . If sufficient samples of R_{b1}^i have been observed during the previous (last) duration (e.g., $Z_{b1}^{i,p} / D > 25\%$) and the average likelihood value is approaching the threshold T_{L1} (e.g., $\bar{L}_{b1}^{i,p} < 0.8T_{L1}$), a new appearance of R_{b1}^i may be observed. In the coming duration, a new appearance model $M_a^c(R_{b1}^i) = \{H_{b1}^{i,c}, T_{b1}^{i,c}\}$ is

obtained from a frame in which $R_i(\mathbf{x})$ looks mostly like R_{b1}^i , i.e., $t_c = \arg \min_{t \in D} \{|L_{b1}^{i,t} - \bar{L}_{b1}^{i,p}|\}$. If the average likelihood values are low in the two consecutive durations, the active appearance model is replaced. First, the new appearance model $M_a^c(R_{b1}^i)$ is compared with the ones in the model base according to (11a). If one is sufficiently close to the new model (i.e. the similarity is larger than $T_{L1} + \varepsilon$), it will be used to replace the active model. Otherwise, the active model will be replaced by the new model. Meanwhile, the new model is also placed into the base. If the model base is already full, the oldest one is then replaced by the new model.

Let T_{b2}^i be the PCR descriptor of the i -th type-2 CBR R_{b2}^i , the *appearance model* of R_{b2}^i is then defined as $M_a(R_{b2}^i) = (T_{b2}^i)$. The *spatial model* of R_{b2}^i describes the range of the homogeneous region in the image. A binary mask $U_{b2}^i(\mathbf{x})$ is used for it, i.e., $M_s(R_{b2}^i) = (U_{b2}^i(\mathbf{x}))$. The spatial model may have to be adjusted in initialization duration when sufficient samples have been observed according to the likelihood values.

Again, a model base is employed to deal with the appearance variations of the type-2 CBRs from day to night. The model base of the i -th type-2 CBR R_{b2}^i is $MB(R_{b2}^i) =$

$\{M_a^k(R_{b2}^i)\}_{k=1}^{K_{b2}^i}, K_{b2}^i \leq K_{b2}^*$. The models in the model base are learned incrementally through the time durations. First, at each time step t , the binary image of observed parts for R_{b2}^i is generated as $V_{b2}^{i,t}(\mathbf{x}) = \mu_{TL2}(L_{b2}^{i,t}(\mathbf{x}))$. The overlapping ratio between the exposed parts and the spatial model for R_{b2}^i at time t is

$$q_{b2}^{i,t} = \frac{V_{b2}^{i,t} \cap U_{b2}^i}{V_{b2}^{i,t} \cup U_{b2}^i} \quad (17a)$$

where ' $\cap$ ' means intersection and ' $\cup$ ' means union. The larger the ratio is, the more parts of R_{b2}^i are exposed and less pixels of other objects would be involved. At the end of each duration, the times of observing the large part of R_{b2}^i during the period is

$$Z_{b2}^{i,c} = \sum_{t \in D} \mu_{TH}(q_{b2}^{i,t}) \quad (18a)$$

and the average similarity value between the observed parts and its active model can be computed as

$$\bar{S}_{b2}^{i,c} = \frac{1}{Z_{b2}^{i,c}} \sum_{t \in D} \log P_L(\bar{T}_{b2}^{i,t} | \bar{T}_{b2}^i) \cdot \mu_{TH}(q_{b2}^{i,t}) \quad (19a)$$

where $P_L(\bar{T}_{b2}^{i,t} | \bar{T}_{b2}^i)$ is calculated according to (10a) with normalized PCRs and $T_H = 75\%$ is used. Like the operation for type-1 CBRs, if sufficient samples have been observed during

the last duration (i.e., $Z_{b2}^{i,p}/D > 25\%$) and the average similarity value is approaching the threshold T_{L2} (e.g., $\bar{S}_{b2}^{i,p} < 0.8T_{L2}$), a new appearance model $M_a^c(R_{b2}^i)$ is generated from the current duration. If the average similarity values are low in the two consecutive durations, the active appearance model will be replaced. If there is a model in the base which is close enough to the new appearance model $M_a^c(R_{b2}^i)$ (i.e. the similarity is larger than $T_{L2} + \varepsilon$), it will be used to replace the active model. Otherwise, the active model will be replaced by the new model. Meanwhile, the new model is also placed into the base. If the model base is already full, the oldest one is then replaced by the new model.

Let $\{R_b^i\}_{i=1}^{Nb}$ be the CBRs of a scene. Given a coming frame $I_t(x)$ and a local region $R_t(x)$ centered at x in $I_t(x)$, the posterior probability of $R_t(x)$ belonging to a CBR R_b^i is

$$P(R_b^i | R_t(x)) = \frac{P(R_t(x) | R_b^i) P(R_b^i | x)}{P(R_t(x))} \quad (20a)$$

The prior probability $P(R_t(x))$ is the same for every pixel in an image. Then the log posterior probability of $R_t(x)$ belonging to R_b^i in the current frame $I_t(x)$ is defined as

$$Q_{b1}^i(R_t(x)) = \log P(R_t(x) | R_b^i) + \log P(R_b^i | x) \quad (21a)$$

The position of a type-1 CBR is already determined by its spatial model. The prior probability $P(R_b^i | x)$ is 1 for the position and 0 otherwise. Then, the log posterior probability is equivalent to the log likelihood at the position, i.e., $Q_{b1}^i = L_{b1}^i(R_t(x_{b1}^i)) = L_{b1}^i$ for R_{b1}^i . A rate of occluded times over recent frames for each type-1 CBR is used. For R_{b1}^i , the rate is computed as

$$r_{b1}^{i,t} = \beta r_{b1}^{i,t-1} + (1 - \beta)(1 - \mu_{T_{L1}}(Q_{b1}^{i,t})) \quad (22a)$$

where β is a smooth factor and $\beta = 0.5$ is chosen. A high rate value (close to 1) indicates that R_{b1}^i has been occluded in recent frames.

From the spatial model $U_{b2}^i(x)$ of the i -th type-2 CBR R_{b2}^i , the prior probability of a pixel x belonging to the region R_{b2}^i can be defined as

$$P(R_{b2}^i | x) = \frac{1}{|R_b(x)|} \sum_{s \in R_b(x)} U_{b2}^i(s) \quad (23a)$$

Combining (21a), (14a), and (23a), the log posterior probability of that x is an exposed point of R_{b2}^i is

$$Q_{b2}^{i,t}(x) = L_{b2}^{i,t}(x) + \log P(R_{b2}^i | x) \quad (24a)$$

A rate of occluded times over recent frames at each pixel for each type-2 CBR is used. First, to be robust to noise and effect of boundaries, an occluded pixel of a type-2 CBR is confirmed on the local neighborhood $R_t(\mathbf{x})$. Let r_1 be the proportion of pixels belonging to R_{b2}^i in the neighborhood region, i.e.,

$$r_1 = \frac{1}{|R_t(\mathbf{x})|} \sum_{s \in R_t(\mathbf{x})} U_{b2}^i(s) \quad (25a)$$

and r_2 be the proportion of exposed pixels of R_{b2}^i in $R_t(\mathbf{x})$ according to the posterior estimates, i.e.,

$$r_2 = \frac{1}{|R_t(\mathbf{x})|} \sum_{s \in R_t(\mathbf{x})} \mu_{T_Q}(Q_{b2}^{i,t}(s)) \quad (26a)$$

where T_Q is chosen as slightly lower than $2T_{L2}$. Then, an occluded pixel of R_{b2}^i is confirmed if the majority of the pixels within its neighborhood are of R_{b2}^i and less of them are observed in the current frame. Now the rate is computed as

$$r_{b2}^{i,t}(\mathbf{x}) = \beta r_{b2}^{i,t-1}(\mathbf{x}) + (1-\beta)[\mu_{T_H}(r_1) \cdot \mu_{T_H}(1-r_2)] \quad (27a)$$

where $T_H = 75\%$ is chosen in the example embodiment.

According to the result a contextual interpretation, three learning rates can be applied at each pixel for different situations in the example embodiment;

Normal learning rate to exposed background pixels with small variations;

Low learning rate to occluded background pixels;

High learning rate to exposed background pixels with significant changes.

An image of control code $C_t(\mathbf{x})$ is used, where the value of $C_t(\mathbf{x})$ is 0, 1, 2, or 3 where the low, normal, or high learning rate is applied respectively at the pixel (here 0 is for normal learning rate for non-context pixels used for display). First, for the pixels not associated with any contextual background region, $C_t(\mathbf{x}) = 0$ is set. The rest of $C_t(\mathbf{x})$ is determined according to the results of contextual interpretation. For a pixel $\mathbf{x}$ within the i -th type-1 CBR R_{b1}^i , if $r_{b1}^{i,t} \geq 0.7$ that means the CBR is being blocked by a foreground object, $C_t(\mathbf{x}) = 1$ is set. Otherwise, if $I_t(\mathbf{x})$ is detected as a background point by background subtraction, $C_t(\mathbf{x}) = 2$ is set since the CBR is exposed and no significant appearance change is found, but if $I_t(\mathbf{x})$ is detected as a foreground point by background subtraction, a high rate should be applied since it is detected as an exposed CBR point with significant appearance change, i.e., $C_t(\mathbf{x}) = 2$. For a pixel of the i -th type-2 CBR R_{b2}^i , if $r_{b2}^{i,t}(\mathbf{x}) \geq 0.7$ that means the patch of the CBR is being occluded by a foreground object, $C_t(\mathbf{x}) = 1$ is set. Otherwise, if $I_t(\mathbf{x})$ is detected as a background point by background subtraction,

$C_t(x) = 2$ is set for exposed part of the type-2 CBR with no significant appearance change. But if $I_t(x)$ is detected as a foreground point by background subtraction, $C_t(x) = 3$ is set for the an exposed neighborhood of the type-2 CBR with significant appearance change.

To smoothen the control code temporally at each pixel, four control code images are used. The first two are the previous and current control code images described above, i.e., $C_{t-1}(x)$ and $C_t(x)$, and the second two images are the control codes which really applied for pixel-level background maintenance, i.e., $C^*_{t-1}(x)$ and $C^*_t(x)$. The applied control code to the current frame at pixel x is determined by the votes from three other control codes $C_{t-1}(x)$, $C_t(x)$, and $C^*_{t-1}(x)$. If at least two of the three codes are the same, the control code of high votes is selected. If the three codes are different from each other, the normal learning rate is used, i.e., $C^*_t(x) = 2$.

To evaluate the effect of context-controlled background maintenance on adaptive background subtraction, the example embodiment was applied to, two existing methods of ABS were implemented. They are the methods based on Mixture of Gaussian (MoG) [4] and Principal Feature Representation (PFR) [2]. Hence, four methods, MoG, Context-Controlled MoG (CC MoG), PFR, and Context-Controlled PFR (CC PFR) were compared. In the test, the normal learning rate of the example embodiment as described above was set to the constant learning rate used for the existing methods of ABS. The high learning rate was set to the double of the normal learning rate and the low learning rate was set to zero. In Figure 1, the leftmost image 102 is a snapshot with manually cropped out contextual background regions e.g. 104, which are type-2 CBRs in this example. In the snapshot image 102, the type-2 CBRs are surrounded by polygon boundaries e.g. 106 of different colors. The second column 108 shows a sample frame from the sequence 110 and the corresponding ground truth 112 of the foreground. The rest of the images in the upper row 114 are: the segmented results by MoG 116, CC MoG (Context-Controlled MoG) 118, and the corresponding control image 120. The three images in the lower row 122 are the segmented results of PFR 124 and CC PFR (Context- Controlled PFR) 126, and the corresponding control image 128. In the control images 120, 128, the black regions e.g. 130 do not belong to any CBR, the gray regions e.g. 132 are exposed parts of the CBRs with no significant appearance changes, and the white regions e.g. 134 are occluded parts of the CBRs.

According to the example embodiment, for pixels in the regions of exposed parts of the CBRs with no significant appearance changes, the normal learning rate is applied, for pixels in regions of occluded parts of the CBRs, the low learning rate is used. For pixels in regions of exposed parts of CBRs with significant changes (not applicable in the scene shown in Figure 1), the high learning rate would be used as described above. The scene in the image 102 is a meeting room with four marked type-2 CBRs for the table surface, the ground surface, wall surfaces, and

the chair. In this sequence of 5250 frames, there was no overstaying objects or overcrowds. However, several people kept e.g. 138 moving around, staying somewhere for a while, and performing various activities. Therefore, the center parts of the scene were frequently occluded by persons. Using a constant learning rate in the unmodified ABS methods, some appearance features of the persons were learned into the background models, and then the background subtraction failed to extract the complete figures of the persons in the incoming frames (see images 116, 124). One example frame, Frame #102810, is displayed in Fig. 1.

By using context-controlled background maintenance of the example embodiment applied to the ABS methods, the persons were segmented satisfactorily (see images 118, 126). A quantitative evaluation on 12 frames sampled from the sequence every 200 frames started from Frame #101410 (empty frames were skipped) is listed in Table 1, where the metric value is defined as the ratio between the intersection and union of the ground truth and the segmented regions. According to [2], the performance is good if the metric value is larger than 0.5 and nearly perfect if the metric value is larger than 0.8. From Table 1, it can be seen that, by using the context-controlled background maintenance of the example embodiment applied to the existing ABS methods, the performance of adaptive background subtraction on situations of complex foreground activities can be improved significantly.

Table 1

	MoG	CC_MoG	PFR	CC_PFR
Frequent Activities	0.2579	0.6678	0.5407	0.8082

The contextual features of the example embodiment capture the global information. Such global information may not always lead to a precise segmentation in position, especially along boundary regions of objects. However, if fed with correct samples continuously, the pixel-level statistical models can be tuned to characterize the background appearance accurately at each pixel. Then the pixel-level background models can be used to preferably achieve a precise segmentation of foreground objects.

The example embodiment exploits contextual interpretation to control the pixel-level background maintenance for adaptive background subtraction. Experimental results show that the example embodiment can improve the performance of adaptive background subtraction for at least situations of high foreground complexities.

Figure 2 shows a flow chart 200 illustrating a method of background updating for adaptive background subtraction in a video signal according to the example embodiment. At step 202, one or more contextual background representation types are defined. At step 204, an image of a scene in the video signal is segmented into foreground and background regions. At step 206, each background region is classified as belonging to one of the contextual background representation types. At step 208, an orientation histogram representation (OHR), a principle colour representation (PCR), or both, are determined of each background region. At step 210, a current image of the scene in the video signal is received. At step 212, it is determined whether respective pixels in image regions of the current image spatially corresponding to the background regions are occluded or exposed. At step 214, different learning rates are set for the adaptive background subtraction for pixels that are occluded and for pixels that are exposed respectively.

While the described example embodiment started from manually cropped out contextual background regions in a snapshot, image segmentation and background object recognition for automatic initialization of contextual models may be performed in different embodiments.

In the example embodiment, principal color representation (PCR) is applied for efficient appearance-based multi-object tracking. In a video surveillance system, object tracking may be applied to a sequence of segmented images generated by background subtraction. In such a case, each segmented image may contain one or several isolated foreground regions. Further, each region may consist of one target object (e.g., a walking person) or a group of target objects (when objects overlap from the camera view point). The example embodiment uses the principal color representation (PCR) for modeling and characterizing the appearance of target objects as well as the segmented regions. For an image sequence captured from a natural public site, each image may contain one or several objects. These objects in the image may overlap on some occasions. Further, the poses, scales, and motion modes of objects can change significantly during the overlap. It has been recognized by the inventors that these issues make the shape-based object tracking a rather challenging task. However, the inventors have recognized that it is much less likely that a target object change colors in a sequence from a surveillance camera. Hence, using global color features of an individual object can provide a relatively stable and constant way for object appearance description. This can also lead to a better discrimination of multiple target objects in the scene.

In video surveillance, an object of interest (e.g., a person, vehicle, luggage, etc.) may render a few dominant colors which only span a small portion of the entire color space. Let the

n th foreground region detected from the frame at time t be $R^n_t(x)$, where $x = (x, y)^T$ denotes the position of a pixel in the region. Then the corresponding principal color representation (PCR) can be defined as

$$T^n_t = \{s_n, \{E^n_t = \{c_n^i, s_n^i\}_{i=1}^N\}\} \quad (1)$$

where s_n is the size of the region (or the total number of the pixels within the region), $c_n^i = (r_n^i, g_n^i, b_n^i)^T$ is the RGB values of the i th most significant color under the original color resolution (i.e., 256 levels for each channel), and s_n^i is the significance of c_n^i for the region. The components E^n_t are sorted in descending order according to the significance values of the principal colors. Let the current frame of input color images be $I_t(x)$, then the significance of i th principal color can be defined as

$$s_n^i = \sum_{x \in R^n_t} \omega(x) \delta(I_t(x), c_n^i) \quad (2)$$

where $\omega(x)$ is a weight function and $\delta(\cdot, \cdot)$ is a delta function. In the example embodiment, $\omega(x) = 1$ is chosen for isolated objects or regions. When locating an object in a group, $\omega(x)$ may not be equal to 1. If necessary, other weight functions can be used, e.g. Gaussian kernel to suppress the noise around the object's boundary [5]. $\delta(c_1, c_2)$ equals to 1 when $c_1 = c_2$, otherwise it is equal 0. However, in the example embodiment a color distance is used which is not sensitive to noise and illumination changes

$$d(c_1, c_2) = 1 - \frac{2 \langle c_1, c_2 \rangle}{\|c_1\|^2 + \|c_2\|^2} \quad (3)$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product. The color distance in (3) is then applied to compute the delta function in (2) as

$$\delta(c_1, c_2) = \begin{cases} 1, & \text{if } d(c_1, c_2) < \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$\varepsilon = 0.005$ is chosen in the example embodiment.

The PCR T^n_t contains the first N significant colors and their statistics for the region $R^n_t(x)$. Since a region of one or a few objects manifests only a few dominant colors, it is possible to find a small number N to approximate the color features of the region, i.e.,

$$\sum_{i=1}^N s_n^i \geq \gamma s_n \text{ with } \gamma = 90\% \quad (5)$$

In the example embodiments, using $N = 50$ in (5) lead to satisfactory results for almost all the regions containing one or a group of objects.

Fig. 3 shows two examples of PCRs where one image 300 contains two isolated individuals and another image 302 contains a group of 5 persons. The PCRs for the foreground regions are generated through scanning the respective regions, and are shown in the histograms 312, 314 respectively. Details of the algorithm for the foreground region $R^n_t(x)$ (see white areas e.g. 304, 306 in the segmented images 308, 310 respectively) are summarized in Table 2.

TABLE 2

THE ALGORITHM TO GENERATE PCR FOR REGION $R^n_t(x)$

1)	$M=0$. Scan the pixels within the bounding box B^n_t of the region $R^n_t(x)$. If $x \in R^n_t(x)$, perform steps 2-4;
2)	From $i=0$ to $M-1$: compute $d(I_i(x), c^n_i)$. If $d(I_i(x), c^n_i) < \varepsilon$, then $s^n_i \leftarrow s^n_i + 1$. Go to 4;
3)	Otherwise, set E^n_M with $c^n_M = I_i(x)$ and $s^n_M = 1$. Set $M \leftarrow M + 1$;
4)	Sorting $\{E^n_i\}_{i=1}^M$ in T^n_t in descending order according to the significance value s^n_i ;
5)	After finishing the scanning, select the first N elements to form the PCR of $R^n_t(x)$

The aim of object tracking is to allocate a tracked object in the coming frame according to its previous appearance. To achieve this, the likelihood, or the conditional probability of observing the tracked object in a region of the current frame, has to be evaluated. In the following, the likelihoods are first defined based on the original and normalized PCRs of the tracked object and a region. This is then extended to the scale-invariant likelihood.

Let O^{m}_{t-1} be the m th tracked object described by its PCR $T^{m}_{t-1} = \{s_m, \{E^i_m = (c^i_m, s^i_m)\}_{i=1}^N\}$ obtained previously when the tracked object was an isolated object, and R^n_t be the n th foreground region detected at time t . According to the law of the total probability, the likelihood of the object O^{m}_{t-1} in the region R^n_t can be defined as

$$P(R^n_t | O^{m}_{t-1}) = \sum_{i=1}^N P(R^n_t | E^i_m) P(E^i_m | O^{m}_{t-1}) \quad (6)$$

where each $P(R^n_t | E^i_m)$ is the likelihood of that the object O^{m}_{t-1} appearing in the region R^n_t based on the partition evidence E^i_m , and $P(E^i_m | O^{m}_{t-1})$ is the conditional probability of the evidence E^i_m given the object O^{m}_{t-1} . Using the PCR T^{m}_{t-1} of the object O^{m}_{t-1} , the conditional probability $P(E^i_m | O^{m}_{t-1})$ can be defined as the weight of the principal color c^i_m for its appearance,

$$P(E_m^i | O_{t-1}^m) = \frac{s_m^i}{s_m} \quad (7)$$

Using the PCRs T_{t-1}^m of the object O_{t-1}^m and T_t^n for the region R_t^n , the likelihood $P(R_t^n | E_m^i)$ can be computed according to their significance values of the same color component c_m^i

$$P(R_t^n | E_m^i) = \min \left\{ 1, \frac{1}{s_m^i} \sum_{j=1}^N \delta(c_m^i, c_n^j) s_n^j \right\} = \frac{1}{s_m^i} \min \left\{ s_m^i, \sum_{j=1}^N \delta(c_m^i, c_n^j) s_n^j \right\} \quad (8)$$

Substituting (7) and (8) into (6) yields

$$P(R_t^n | O_{t-1}^m) = \frac{1}{s_m} \sum_{i=1}^N \min \left\{ s_m^i, \sum_{j=1}^N \delta(c_m^i, c_n^j) s_n^j \right\} \quad (9)$$

It is noted that the above likelihood (9) is evaluated under the assumption that the size variation of the object is small. However, if the size change is large the likelihood value will be affected. Therefore, a definition of likelihood based on the normalized PCRs is used in the example embodiment. Let

$\tilde{T}_{t-1}^m = \{1, \{\tilde{E}_m^i = (c_m^i, \tilde{s}_m^i)\}_{i=1}^N\}$ be the normalized PCR of the object O_{t-1}^m , where $\tilde{s}_m^i = s_m^i / s_m$, and $\tilde{T}_t^n = \{1, \{\tilde{E}_n^j = (c_n^j, \tilde{s}_n^j)\}_{j=1}^N\}$ be the normalized PCR of the region R_t^n with $\tilde{s}_n^j = s_n^j / s_n$. The likelihood based on the normalized PCRs becomes

$$\tilde{P}(R_t^n | O_{t-1}^m) = \sum_{i=1}^N \min \left\{ \tilde{s}_m^i, \sum_{j=1}^N \delta(c_m^i, c_n^j) \tilde{s}_n^j \right\} \quad (10)$$

If the region R_t^n only contains a single object, the likelihood based on the normalized PCRs is more accurate than that based on the original PCRs. However, if O_{t-1}^m is one of the objects in the group R_t^n , the likelihood on original PCRs is better. Hence, the scale-invariant likelihood of observing a given object O_{t-1}^m in the region R_t^n is defined as,

$$P^*(R_t^n | O_{t-1}^m) = \max\{P(R_t^n | O_{t-1}^m), \tilde{P}(R_t^n | O_{t-1}^m)\} \quad (11)$$

Heuristically, if the region R_t^n is a single object, the likelihood computed from normalized PCRs appears more reliable. However, if O_{t-1}^m is one of the objects in a group R_t^n , the likelihood from un-normalized PCRs appears better since the object is smaller than the group. Equ (11) can provide a suitable measurement for these two cases in the example embodiment.

Object tracking in video surveillance aims at maintaining a unique identification (*id*) for each target object and providing its time-dependent positions in the scene. When multiple target objects frequently merge and separate from one another in a public site, tracking one individual object is no longer an independent process. Multi-object tracking can be formulated as a global Maximum A Posterior (MAP) problem for all the tracked objects. With the segmented foreground regions provided by background subtraction, in the example embodiment the global MAP problem can be approximately decomposed as two subproblems: assignment and location. Using the principal color representation (PCR) and the associated likelihood function, the example embodiments uses sequential solutions to these two subproblems, as detailed below.

Let $\mathcal{O}_{t-1} = \{O_{t-1}^m\}_{m=1}^{M_{t-1}}$ be the set of tracked target objects in the previous frame $I_{t-1}(x)$, and $\Theta_{t-1} = \{\theta_{t-1}^m\}_{m=1}^{M_{t-1}}$ be the set of state parameters describing their positions at time $t-1$. The task of multi-object tracking is to estimate the states Θ_t of tracked objects in the current frame $I_t(x)$ given their previous appearance models $\mathcal{O}_{t-1}$ and states Θ_{t-1} . This can be formulated as the Maximum A Posterior (MAP) estimation for the state parameters Θ_t^* .

$$\Theta_t^* = \arg \max_{\Theta_t} P(\Theta_t | I_t(x), \mathcal{O}_{t-1}, \Theta_{t-1}) \quad (12)$$

When several objects overlap one another, they cannot be tracked as independent objects. With foreground regions $\mathcal{R}_t = \{R_t^k\}_{k=1}^{N_t}$ provided by background subtraction, (12) can be simplified as

$$\Theta_t^* = \arg \max_{\Theta_t} P(\Theta_t | \mathcal{R}_t, \mathcal{O}_{t-1}, \Theta_{t-1}) \quad (13)$$

where the tracked objects $\{O_{t-1}^m\}_{m=1}^{M_{t-1}}$ are in the regions $\{R_{t-1}^j\}_{j=1}^{N_{t-1}}$ from the previous frame. If a region (e.g., R_{t-1}^j) only contains one object (e.g., O_{t-1}^m) it is an isolated object, otherwise the region is a group. Objects belonging to different regions in the previous frame may merge into a new group region (e.g., R_t^k) in the current frame. Also, the objects in a group region (e.g., R_{t-1}^j) in the previous frame may separate to several regions in the current frame. For real-time processing with moderate or high frame rate of image acquisition, the inter-frame movements of target objects are usually small. This implies that there is always an overlap between the regions of the same object in the consecutive frames. Exploiting such a relation, the problem (13) can be further decomposed by using *directed acyclic graphs* (DAGs).

The directed acyclic graphs (DAGs) for the regions detected in the consecutive frames $I_{t-1}(x)$ and $I_t(x)$ are constructed in the following way. Let the regions from the previous and current frames be denoted as the nodes and be laid in two layers: the parent layer and the child layer. The parent layer consists of nodes representing the regions $\{R_{t-1}^j\}_{j=1}^{N_{t-1}}$ in the previous frame $I_{t-1}(x)$, and the child layer consists of nodes denoting the regions $\{R_t^k\}_{k=1}^{N_t}$ in the current frame $I_t(x)$. Suppose R_{t-1}^j and R_t^k are the j th and k th regions in the previous and current frames, respectively, then the directional link from R_{t-1}^j to R_t^k can be defined as

$$l_{jk} = \begin{cases} 1, & \text{if } (R_{t-1}^j \cap R_t^k) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

This implies that there is a link only when the two regions have some overlap. A directed acyclic graph (DAG) is formed by a set of nodes in which every node connects to one or more nodes in the same group. A set of DAGs (graphs) can be generated. An example of graphs for two consecutive frames is illustrated in Figure 4. The notations for the DAGs are defined as follows. For the i th graph, the parent nodes are denoted as $\{n_0^{i,p}\}_{p=1}^{N_{0,i}}$ where each node $n_0^{i,p}$ represents one of the regions $\{R_{t-1}^j\}_{j=1}^{N_{t-1}}$, and the child nodes are denoted as $\{n_1^{i,q}\}_{q=1}^{N_{1,i}}$ where each node $n_1^{i,q}$ represents one of the regions $\{R_t^k\}_{k=1}^{N_t}$. The i th DAG can thus be denoted as $G_i = (\{n_0^{i,p}\}_{p=1}^{N_{0,i}}, \{n_1^{i,q}\}_{q=1}^{N_{1,i}}, \{l_{pq}^i\})$. The objects in a parent node $n_0^{i,p}$ are denoted as $\{o_{t-1}^{m,p}\}_{m=1}^{M_{i,p}}$. If $M_{i,p} = 1$ then the node is a single object, otherwise it is a group of $M_{i,p}$ objects. The object $o_{t-1}^{m,p}$ is one of the objects $\{O_{t-1}^m\}_{m=1}^{M_{t-1}}$. The objects in a child node $n_1^{i,q}$, which may be newly generated objects or objects tracked from its parent nodes, are denoted as $\{o_t^{n,q}\}_{n=1}^{M_q}$. After processing all graphs, the objects in the child nodes are reordered as $\{O_t^n\}_{n=1}^{M_t}$. They are the set of tracked target objects in the current frame.

Since there is no link between different DAGs, the objects in the parent nodes of one graph can be tracked independently of the other graphs. Let O_{t-1}^i represent the set of objects in the parent nodes of graph G_i , i.e., $O_{t-1}^i = \{\{o_{t-1}^{m,p}\}_{m=1}^{M_{i,p}}\}_{p=1}^{N_{0,i}}$. Then the probability of the states for all the tracked objects in image $I_t(x)$ becomes

$$P(\Theta_t | \mathcal{R}_t, O_{t-1}, \Theta_{t-1}) = \prod_{i=1}^{L_G} P(\Theta_t^i | G_i, O_{t-1}^i, \Theta_{t-1}^i) \quad (15)$$

where $\Theta_t = (\Theta_t^1, \dots, \Theta_t^{L_G})$ and L_G is the number of DAGs. According to (15), (13) can be decomposed as finding a $(\Theta_t^i)^*$ for each graph such that

$$(\Theta_t^i)^* = \arg \max_{\Theta_t^i} P(\Theta_t^i | G_i, O_{t-1}^i, \Theta_{t-1}^i) \quad (16)$$

If there are several parent and child nodes in a graph, and some parent nodes represent groups, (16) is still a nontrivial problem. The example embodiment solves the problem in two sequential steps from coarse to fine. The problem is decomposed approximately as two sub-problems: *assignment* and *location*. The coarse assignment process assigns each object in a parent node to one of its child nodes while the fine location process determines the new states of the objects assigned to each child node.

Assignment:

In this step, the tracked objects in each parent node is assigned to its child nodes based on the largest posterior probability. Let $\hat{\Theta}_t^i$ be the parameter vector describing the assignment of the tracked objects $\mathcal{O}_{t-1}^i$ to the child nodes $\{n_1^{i,q}\}_{q=1}^{N_{1,t}^i}$. The posterior probability of the assignment for graph G_t can be expressed as

$$P(\hat{\Theta}_t^i | G_t, \mathcal{O}_{t-1}^i, \Theta_{t-1}^i) = \prod_{p=1}^{N_{0,t}} P(\hat{\Theta}_t^{i,p} | \mathcal{N}_1^{i,c(p)}, \mathcal{O}_{t-1}^{i,p}, \Theta_{t-1}^{i,p}) \quad (17)$$

where $\mathcal{O}_{t-1}^{i,p} = \{o_{t-1}^{m_{i,p}}\}_{m_{i,p}=1}^{M_{i,p}}$ are the tracked objects in node $n_0^{i,p}$, and $\mathcal{N}_1^{i,c(p)} = \{n_1^{i,q} : i_{pq}^i = 1\}_{q=1}^{N_{1,t}^i}$ are the child nodes of $n_0^{i,p}$. The parameter vector is $\hat{\Theta}_t^i = (\hat{\Theta}_t^{i,1}, \dots, \hat{\Theta}_t^{i,N_{0,t}^i})$. The best assignment for the tracked objects $\{o_{t-1}^{m_{i,p}}\}_{m_{i,p}=1}^{M_{i,p}}$ in $n_0^{i,p}$ is chosen such that it results in the best observation of the objects in the corresponding child nodes, that is

$$(\hat{\Theta}_t^{i,p})^* = \arg \max_{\hat{\Theta}_t^{i,p}} P(\hat{\Theta}_t^{i,p} | \mathcal{N}_1^{i,c(p)}, \mathcal{O}_{t-1}^{i,p}, \Theta_{t-1}^{i,p}) \quad (18)$$

Here $\hat{\Theta}_t^{i,p}$ can be considered as the coarse tracking parameters indicating in which child nodes (regions) the objects are observed without concerning the exact new positions of the tracked objects in the child regions.

Location:

In this step, the new states of the tracked objects assigned to each child node (e.g., region R_t^k) are determined. Let $\mathcal{O}_t^{i,q} = \{o_t^{n_{i,q}}\}_{n_{i,q}=1}^{N_{i,q}}$ be the objects assigned to the child node $n_1^{i,q}$ from its parent nodes. That is, $\mathcal{O}_t^{i,q}$ is a subset of $\mathcal{O}_{t-1}^i$ according to the assignment parameters $(\hat{\Theta}_t^i)^* = ((\hat{\Theta}_t^{i,1})^*, \dots, (\hat{\Theta}_t^{i,N_{0,t}^i})^*)$. After the assignment, objects in each child node can be tracked independently of objects in the other child nodes. Hence, the posterior probability of the new states for the tracked objects in the graph G_t can be evaluated as

$$P(\Theta_t^i | G_t, \mathcal{O}_{t-1}^i, \Theta_{t-1}^i) = \prod_{q=1}^{N_{1,t}^i} P(\Theta_t^{i,q} | n_1^{i,q}, \mathcal{O}_t^{i,q}, \Theta_{t-1}^{i,q}) \quad (19)$$

where $\Theta_t^i = (\Theta_t^{i,1}, \dots, \Theta_t^{i,N_{i,t}})$, and $\Theta_{t-1}^{i,q}$ is the set of previous state parameters for objects $O_t^{i,q}$ in $n_1^{i,q}$. From (19), locating the objects in the child node $n_1^{i,q}$ is expressed as

$$(\Theta_t^{i,q})^* = \arg \max_{\Theta_t^{i,q}} P(\Theta_t^{i,q} | n_1^{i,q}, O_t^{i,q}, \Theta_{t-1}^{i,q}) \quad (20)$$

Multi-object tracking thus becomes finding the solutions for Eqs. (18) and (20) in the example embodiment. Further sequential solutions for (18) and (20) based on PCR are used and described below.

Assuming that $\{R_t^k\}_{k=1}^{N_t}$ are the foreground regions and $\{B_t^k\}_{k=1}^{N_t}$ are their bounding boxes detected at time t , then their PCRs can be obtained as $\{T_t^k\}_{k=1}^{N_t}$. Let $\{G_t\}_{t=1}^{L_G}$ be the set of directed acyclic graphs (DAGs) for the foreground regions between the consecutive frames. If there is only one object in a graph G_t , then the object will be tracked as an isolated object. Otherwise, multi-object tracking will be performed according to Eqs. (18) and (20). For tracking multiple objects in a group, the posterior probability of the new state for each object is determined on both spatial position and depth relationship. Hence, $2\frac{1}{2}D$ state is used for each object. The state vector for an object O_t^n is $\theta_t^n = (b_t^n, v_t^n)$ where b_t^n is the bounding box describing its spatial position and v_t^n is the likelihood value describing its depth position in the group.

Tracking Isolated Objects

If the i th graph consists of only one child node (i.e., $G_i = \langle n_1^{i,1} \rangle$), a new object appears and is initialized as o_t^1 in G_i with a new *id* number. Suppose the node $n_1^{i,1}$ represents the region R_t^k , then the PCR and bounding box of o_t^1 are set as $T_t^1 = T_t^k$ and $b_t^1 = B_t^k$. Since o_t^1 is an isolated object, it is not occluded by any other objects. The depth state is set as $v_t^1 = P(R_t^k | o_t^1) = 1$. Its $2\frac{1}{2}D$ state parameter vector is $\theta_t^1 = (b_t^1, v_t^1)$.

If the i th graph contains one parent node and one child node, and the parent node is associated with one object, the graph represents the simple case of isolated object tracking. Let the graph be $G_i = \langle n_0^1, n_1^1, l_{11}^1 \rangle$, the object in the parent node be O_{t-1}^m , and the child node n_1^1 represent the region R_t^k , then the object O_{t-1}^m is updated as o_t^1 in n_1^1 (i.e., O_{t-1}^m and o_t^1 have the same *id* number). Its state becomes $\theta_t^1 = (b_t^1, v_t^1)$ with $b_t^1 = B_t^k$ and $v_t^1 = 1$. In addition, its PCR is updated as $T_t^1 = T_t^k$ to follow the gradual variation of the object.

If the i th graph only contains one parent node which has no child nodes, then the previous objects in the parent node are assumed to have disappeared in the current frame. Tracking is terminated for these objects.

Tracking Multiple Objects in a Graph

If the i th graph G_i contains multiple parent nodes or child nodes, the operations of *assignment* and *location* will be performed. In the following description of the operations for one graph, for the sake of notational convenience the index i for the graph G_i is omitted below.

Assignment:

Let n_0^p be a parent node in the graph G , $\mathcal{O}_{t-1}^p = \{o_{t-1}^m\}_{m=1}^{M_p}$ be the associated objects, and $\mathcal{N}_1^{c(p)} = \{n_1^q : l_{pq} = 1\}_{q=1}^{N_1}$ be its child nodes. If the parent node has more than one child node, the assignment of objects $\mathcal{O}_{t-1}^p$ is determined by Eq. (18). However, with varying numbers of objects and child nodes, Eq. (18) is a nontrivial problem of optimal configuration. To make the problem tractable, a sequential solution is proposed based on their PCRs and the depth relations among the objects.

In each group, the close and non-occluded objects have richer visible information than the distant or occluded objects. This means that an occluded object less affects the tracking of the objects occluding it. Hence, the assignment can be solved sequentially from the most visible one to the least visible one. Let the objects $\{o_{t-1}^m\}_{m=1}^{M_p}$ in the parent node n_0^p be ordered according to their visible sizes. Assuming that the correct assignment of the object o_{t-1}^m is $\hat{\theta}_t^m = q_m$ which assigns o_{t-1}^m to the child node $n_1^{q_m}$ ($n_1^{q_m} \in \mathcal{N}_1^{c(p)}$), and the child node $n_1^{q_m}$ represents the region $R_t^{k_m}$, then the posterior probability of assignment for objects $\mathcal{O}_{t-1}^p = \{o_{t-1}^m\}_{m=1}^{M_p}$ is computed as

$$P(\hat{\Theta}_t^p | \mathcal{N}_1^{c(p)}, \mathcal{O}_{t-1}^p, \Theta_{t-1}^p) = \prod_{m=1}^{M_p} P(\hat{\theta}_t^m = q_m | \tilde{R}_t^{k_m}(m-1), o_{t-1}^m, \theta_{t-1}^m) \quad (21)$$

where $\tilde{R}_t^{k_m}(m-1) = R_t^{k_m} - \sum_{l=1}^{m-1} \delta(\hat{\theta}_t^l = q_m) o_{t-1}^l$ represents the region after excluding the objects previously assigned to it before o_{t-1}^m . Note that the assignments of the objects $\{o_{t-1}^m\}$ are not independent. The assignment of one object is affected by the previous objects with higher ranks. This means the assignment of each object can be performed one-by-one sequentially from the most to the least visible ones. For each object, the posterior probability of assignment can be evaluated using Bayes rule,

$$P(\hat{\theta}_t^m = q_m | \tilde{R}_t^{k_m}(m-1), o_{t-1}^m, \theta_{t-1}^m) = P(\tilde{R}_t^{k_m}(m-1) | o_{t-1}^m, \hat{\theta}_t^m = q_m) P(\theta_t^m = q_m | \theta_{t-1}^m) \quad (22)$$

The first term on the righthand side of (22) is the likelihood of observing the object o_{t-1}^m in the region R_t^{km} with the exclusion of previously assigned objects, while the second term is the prior probability of $\hat{\theta}_t^m = q_m$ given the previous state θ_{t-1}^m . For assignment, (22) can be evaluated on PCR. Assuming that a child node $n_1^q \in \mathcal{N}_1^{c(p)}$ represents the region R_t^k . Let T_t^k and T_{t-1}^m be the PCRs of R_t^k and o_{t-1}^m , respectively. Using Eqs. (21) and (22), the best assignment of the objects $\{o_{t-1}^m\}_{m=1}^{M_p}$ in the parent node n_0^p can be achieved one-by-one sequentially according to their depth order by

$$(\hat{\theta}_t^m)^* = q_m = \arg \max_{q: n_1^q \in \mathcal{N}_1^{c(p)}} P(\tilde{T}_t^k(m-1) | T_{t-1}^m, \hat{\theta}_t^m = q) P(\hat{\theta}_t^m = q | \theta_{t-1}^m) \quad (23)$$

from $m = 1$ to M_p , where $\tilde{T}_t^k(m-1) = T_t^k - \sum_{l=1}^{m-1} \delta(\hat{\theta}_t^l = q) T_{t-1}^l$ represents the PCR of $\tilde{R}_t^k(m-1) = R_t^k - \sum_{l=1}^{m-1} \delta(\hat{\theta}_t^l = q) o_{t-1}^l$.

The sequential solution to Eq. (18) using Eq. (23) is computed in two steps. First, the objects $\{o_{t-1}^m\}_{m=1}^{M_p}$ in the parent node n_0^p are sorted in a list according to their visible parts. An iterative process is then performed from the most visible to the least visible object. In each iteration, the object in the top of the list is assigned to one of the child nodes according to (23). Once an object is assigned to a child node, it is removed from the list and its visual evidence is excluded from the PCR of the child region. Details are described below.

Let $\{v_{t-1}^m\}_{m=1}^{M_p}$ be the likelihoods of objects $\{o_{t-1}^m\}_{m=1}^{M_p}$ in n_0^p computed in the previous time step. Let s_m be the size of o_{t-1}^m from its PCR T_{t-1}^m . The visible part of o_{t-1}^m in n_1^q is estimated as $\zeta_{t-1}^m = v_{t-1}^m s_m$. We denote $\{\zeta_{t-1}^m\}_{m=1}^{M_p}$ as being sorted in descendent order according to the values of $\{\zeta_{t-1}^m\}_{m=1}^{M_p}$ and placed in a list. Before the iteration, $\tilde{T}_t^k(0) = T_t^k$ is set for the regions associated with the child nodes $\mathcal{N}_1^{c(p)}$.

In each iteration, the top object in the list is pop out. Suppose it is the m th object o_{t-1}^m . The likelihood in (23) is calculated as $P^*(\tilde{T}_t^k(m-1) | T_{t-1}^m)$ according to Eq. (11). The prior probability in (23) is evaluated based on the shape similarity and center distance between the bounding boxes. Let b_{t-1}^m and B_t^k be the bounding boxes of o_{t-1}^m and R_t^k associated with the child node n_1^q , x_c^m and x_c^k be their centers, and d_m and d_k be their diagonal lengths, respectively. Then, the shape similarity between two boxes (center aligned) is defined as $m_s = \frac{b_{t-1}^m \cap B_t^k}{b_{t-1}^m \cup B_t^k}$, where “ $\cap$ ” denotes the intersection and “ $\cup$ ” denotes the union. The center distance between two boxes is defined as $m_d = e^{-\frac{\|x_c^m - x_c^k\|^2}{2d^2}}$ with $d = (d_m + d_k)/2$. The prior probability is defined as $P(\hat{\theta}_t^m = q | \theta_{t-1}^m) = 0.5(m_s + m_d)$. The object o_{t-1}^m is assigned to the child node $n_1^{q_m}$ according to Eq. (23).

The last operation in this iteration is exclusion that removes the visual information of the object α_{t-1}^m from $\tilde{T}_t^{km}(m-1)$ associated with the child node n_1^q . Let T_{t-1}^m and $\tilde{T}_t^{km}(m-1)$ be the PCRs of α_{t-1}^m and $\tilde{R}_t^{km}(m-1)$. The PCR $\tilde{T}_t^{km}(m) = \tilde{T}_t^{km}(m-1) - T_{t-1}^m$ is updated from $\tilde{T}_t^{km}(m-1)$ for its principal colors one by one. For the j th element of principal color c_{km}^j and significance s_{km}^j , the following updating is performed.

$$\begin{aligned}\Delta s &= \min(s_{km}^j, \sum_{i=1}^N \delta(c_{km}^j, c_m^i) s_m^i) \\ s_{km}^j &= s_{km}^j - \Delta s\end{aligned}\quad (24)$$

When all the elements in $\tilde{T}_t^{km}(m-1)$ are updated, the generated PCR is $\tilde{T}_t^{km}(m)$. For the regions associated with the rest of child nodes in $\mathcal{N}_1^{c(p)}$, $\tilde{T}_t^k(m) = \tilde{T}_t^k(m-1)$ is set.

Location:

Let $\mathcal{O}_t^q = \{\alpha_t^n\}_{n=1}^{M_q}$ are the objects assigned to a child node n_1^q in the graph G . The new states of the objects should be determined by solving Eq. (20).

Locating the objects in a region are not independent of each other, but the front ones with richer visible information are less affected by the occluded ones. Hence, in the example embodiment objects in the node are located one by one from the most visible to the least visible ones based on their visible parts. The posterior probability of new states for all the objects in the node can be expressed as

$$P(\Theta_t^q | n_1^q, \mathcal{O}_t^q, \Theta_{t-1}^q) = \prod_{n=1}^{M_q} P(\theta_t^n | \tilde{R}_t^k(n-1), \alpha_{t-1}^n, \theta_{t-1}^n) \quad (25)$$

where $\Theta_t^q = (\theta_t^1, \dots, \theta_t^{M_q})$, R_t^k is the region associated with the node n_1^q , and $\tilde{R}_t^k(n-1) = R_t^k - \sum_{i=1}^{n-1} \alpha_i^t$ represents the region in which the visual evidence of the first $n-1$ objects $(\alpha_t^1, \dots, \alpha_t^{n-1})$ have been excluded at the located positions. According to (25), locating objects $\mathcal{O}_t^q$ according to (20) is equivalent to locating them one by one sequentially according to

$$(\theta_t^n)^* = \arg \max_{\theta_t^n} P(\theta_t^n | \tilde{R}_t^k(n-1), \alpha_{t-1}^n, \theta_{t-1}^n) \quad (26)$$

where $\{\alpha_t^n\}_{n=1}^{M_q}$ are sorted in descendent order according to their visible sizes.

The sequential solution to the problem Eq. (20) and Eq. (26) contains two steps. In the first step, the visible parts of the objects in the node are estimated, and the objects are sorted according to their visible sizes. In the second step, an iterative process is applied to locate the objects one-by-one in the region with a mean-shift algorithm based on PCR. When an object is

located, its visual evidence is excluded from its position in the region. The details are described in the following.

Assuming that an object o_t^n in the child node n_t^q is the object $o_{t-1}^{m_p}$ assigned from the parent n_0^p , then there is $o_{t-1}^n = o_{t-1}^{m_p}$. Let $\{v_{t-1}^n\}_{n=1}^{M_q}$ be the likelihoods of $\{o_{t-1}^n\}_{n=1}^{M_q}$ computed from the previous frame. The likelihood of observing object o_t^n in the child node n_t^q (or region R_t^k) in the current frame can be evaluated as $P^*(R_t^k|o_t^n)$ according to Eq. (11). Since the motion of object between the two consecutive frames is assumed small, the visible parts of the object o_t^n in region R_t^k can be estimated as

$$\zeta_t^n = (\eta v_{t-1}^n + (1 - \eta) P^*(R_t^k|o_t^n)) \cdot \min(s_n, s_k) \quad (27)$$

where s_n and s_k are the sizes of the object o_t^n and the region R_t^k , respectively. In Eq. (27), η is a weight to smooth the estimates from consecutive frames ($\eta = 0.5$ is chosen in this study). $\{o_t^n\}_{n=1}^{M_q}$ are then sorted in descendent order according to the values of $\{\zeta_t^n\}_{n=1}^{M_q}$ and then placed in a list. To perform exclusion for $\tilde{R}_t^k(n-1) = R_t^k - \sum_{l=1}^{n-1} o_t^l$, a weight image $\omega_n(x)$ is used. If the pixel x is likely to belong to one of the previously located objects $(o_t^1, \dots, o_t^{n-1})$, $\omega_{n-1}(x)$ is low (≈ 0), otherwise, it is high (≈ 1). For initialization, set $\omega_0(x) = 1$ for all the pixels belonging to the region R_t^k , and $\omega_0(x) = 0$ otherwise.

In each iteration, the top object in the list is pop up. Assuming that it is the n th object o_t^n with the initial position represented by the previous bounding box $B_n^{(0)} = b_{t-1}^n$ centered at $x_n^{(0)}$, and its PCR is $T_t^n = \{s_n, \{E_n^j = (c_n^j, s_n^j)\}_{j=1}^N\} = T_{t-1}^{m_p}$ coming from the object $o_{t-1}^{m_p}$. Locating

the object o_t^n in the region R_t^k according to Eq. (26) is equivalent to finding a position where the maximum value of probability density occurs for observing the object. This density maximum can be found by employing a mean-shift procedure with a weight mode which can reveal the probability density of observing the object in the neighborhood [5], [6], [7]. A two-stage mean-shift procedure is proposed based on the evidence of the object's principal colors. In the first stage, the gravity center of the pixels of each principal color component is computed as

$$x_n^j = \frac{\sum_{x \in B_n^{(r)}} x \omega_{n-1}(x) \delta(I(x, t), c_n^j)}{\sum_{x \in B_n^{(r)}} \omega_{n-1}(x) \delta(I(x, t), c_n^j)} \quad (28)$$

with $j = 1, \dots, N$, where r indicates a current step of mean-shift iteration.

In the second stage, the new position of the object o_t^n is generated as the weighted average of the gravity centers

$$x_n^{(r+1)} = \frac{\sum_{j=1}^N x_n^j w_j}{\sum_{j=1}^N w_j} \quad (29)$$

where the weight of evidence from principal color c_n^j is defined as the backprojection $w_j = \min(1, s_n^j/s_r^j)$ with $s_r^j = \sum_{x \in B_n^{(r)}} \omega_{n-1}(x) \delta(I(x, t), c_n^j)$. The weight w_j implies that if $s_n^j < s_r^j$ then only s_n^j/s_r^j proportion of the pixels with color c_n^j in the bounding box $B_n^{(r)}$ belong to the object o_t^n , otherwise all the pixels of color c_n^j belong to the object. The mean-shift procedure is terminated once $\|x_n^{(r+1)} - x_n^{(r)}\| < 2$ is satisfied, or the maximum number of iterations is reached (6 in the example). The new location of object o_t^n is the bounding box $b_t^n = B_n^{(r+1)}$ centered at $x_n^{(r+1)}$. Let $\tilde{T}_b(n-1) = \{s_b, \{E_b^n = (c_b^j, s_b^j)\}_{j=1}^N\}$ be the PCR of the part of region $\tilde{R}_t^k(n-1)$ within the bounding box b_t^n , where s_b is the size of the part and the significance is $s_b^j = \sum_{x \in b_t^n} \omega_{n-1}(x) \delta(I(x, t), c_b^j)$. The likelihood of the object o_t^n in the group can be estimated as $v_t^n = P^*(\tilde{T}_b(n-1)|T_t^n)$ according to Eq. (11). The new state parameter of the object o_t^n is then obtained as $(\theta_t^n)^* = (b_t^n, v_t^n)$.

The last operation of the iteration is exclusion which removes the visual information of the tracked object o_t^n from the region R_t^k at its position, or obtaining $\tilde{R}_t^k(n) = R_t^k - \sum_{i=1}^n o_i^l = \tilde{R}_t^k(n-1) - o_t^n$. This operation is done by updating $\omega_{n-1}(x)$ to $\omega_n(x)$. For a pixel $x \in b_t^n$, $\omega_{n-1}(x) > 0$ and $c_x = I_x(x)$, we can obtain the significance of c_x in both T_t^n and $\tilde{T}_b(n-1)$ as

$$s_n(c_x) = \sum_{i=1}^N \delta(c_x, c_n^i) s_n^i \quad \text{and} \quad s_b(c_x) = \sum_{j=1}^N \delta(c_x, c_b^j) s_b^j \quad (30)$$

Let us define $\Delta\omega(x) = \min(1, s_n(c_x)/s_b(c_x))$. The value of $\Delta\omega(x)$ can be considered as the probability of x belonging to the object o_t^n . Hence, the updating of the weight image can be performed as

$$\omega_n(x) = \max\{0, \omega_{n-1}(x) - \Delta\omega(x)\} \quad (31)$$

The complete algorithm of multi-object tracking based on PCR in the example embodiment is summarized in Table 3.

TABLE 3

THE SUMMARY OF THE MULTI-OBJECT TRACKING ALGORITHM

Input: color image $I_t(x)$ and segmented image $S_t(x)$;

Preprocessing: generate graphs $\{G_i, i = 1, \dots, L_G\}$;

For $G_i, i = 1, \dots, L_G$, *do:*

Assignment: for each parent node $n_0^{i,p}$ in G_b , $p = 1, \dots, M_{i,p}$, do:

a.1: if $n_0^{i,p}$ has no child node, the objects in it are deleted;

a.2: if $n_0^{i,p}$ has only one child node $n_1^{i,q}$, all the objects in it are assigned to $n_1^{i,q}$;

a.3: if $n_0^{i,p}$ has multiple child nodes;

a.3.1: sort the objects $\{o_{t-1}^m\}_{m=1}^{M_p}$ in $n_0^{i,p}$, and then assign them one-by-one from the first to the last as follows:

a.3.1.1: assign o_{t-1}^m to the child node $n_1^{i,qm}$ according to (23);

a.3.1.2: exclude the visual information of o_{t-1}^m from the PCR of $n_1^{i,qm}$

Location: for each child node $n_1^{i,p}$ in G_b , $q = 1, \dots, M_q$, do:

l.1: if no object is assigned to the node, check if it is a disappeared object; if not, set it as a new object;

l.2: if only one object is assigned to the node, update the state and PCR of the object;

l.3: if multiple objects are assigned to the node:

l.3.1: sort the objects $\{o_t^n\}_{n=1}^{M_p}$ in the node using (27), and then locate the objects one-by-one as follows:

l.3.1.1: apply mean-shift to locate o_t^n using (28) and (29);

l.3.1.2: exclude the visual evidence of o_t^n at the location in R_t^k using (31)

l.3.2: if the likelihood of observing an object in the region is less than 0.1, set the object as disappeared.

Clearance: if an object disappeared for more than 50 frames, delete the object.

End

The algorithm in the example embodiment includes two phases of processing for each DAG (Directed Acyclic Graph): assignment and location. In the assignment phase, each parent node in the DAG is processed. In the location phase, assigned objects in each child node are tracked. To be robust to the separation of small parts from the tracked object due to segmentation errors, small objects in a group with likelihood values less than 0.1 are set as disappeared. To prevent the losing of small or heavily occluded objects in a group, the records of disappeared objects are kept for 50 frames. When a new object is detected, it is compared with disappeared

objects according to their PCRs, sizes and distances. If it compares to a disappeared object the tracking will be restored, otherwise a new object is created.

In the example embodiment, segmenting individual persons in a group with domain knowledge will be preferred. For example, in the example embodiment knowledge about the sizes and aspect ratios of persons in the scene is used to adapt to segmentation errors.

Figure 5 shows a flow chart 500 illustrating a method of multi-object tracking in a video signal in the example embodiment. At step 502, first and second segmented images of two consecutive frames of the video signal respectively are received, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked. At step 504, one or more directed acyclic graphs (DAGs) are generated for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node. At step 506, for each parent node having two or more child nodes,

- a) the corresponding objects of the foreground region contributing to said each parent node are sorted according to estimated depth in said first image;

- b) the corresponding object having the lowest depth is assigned to one of the child nodes of said each parent node;

- c) a visual content of the assigned corresponding object is removed from the visual data associated with said one child node; and

steps b) to c) are iterated in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes.

At step 508, for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image.

At step 510, for each child node having two or more corresponding objects assigned thereto,

- d) the corresponding objects are sorted according to estimated depth in said each child node in said second image;

- e) a means-shift calculation is applied to locate the corresponding object having the lowest depth in said each child node;

- f) the state and the visual content of the located corresponding object are updated based on the second image;

- g) the updated visual content of the located corresponding object is removed from the visual data associated with said each child node; and

steps e) to g) are iterated in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

When an object stops moving and stays in the same position in the scene for a while, the object would be absorbed into the background gradually with existing background updating techniques. That means the object would be lost in the segmented foreground images. On the other hand, in e.g. crowd scenes, if one can separate the moving objects from the stationary objects in the scene, one can reduce the overlapping of multiple foreground objects. This would make the tracking of each individual easier and more robust. In the described example embodiment, a layer tracking algorithm is designed to track stationary objects through even frequent occlusions. When the object starts moving, the object is identified as moving object and tracked by a moving object tracking algorithm. In the example embodiment, the stationary objects include not only static non-living objects but also include motionless living objects, e.g. a standing or sitting person. Since the living objects may move again, the switching between moving object tracking and stationary object tracking for the target object is preferably smooth with no change of identity in the example embodiment.

When an object stops moving and stays in a scene in frame of a video signal, the appearance variation of the object is typically small through a sequence of frames. A template image of the object is used to represent such a stationary object in the example embodiment.

Let $\{B_i\}_{t=t-\tau_b}^t$ be a sequence of bounding boxes of the i th tracked object in the τ_b most current frames as tracked by a moving object tracking algorithm. If the object has stopped moving, the bounding boxes will overlap each other. For a selected length parameter τ_b , if the spatial intersection of all the boxes is not empty, the object is detected as a stationary object in the example embodiment. In the example embodiment, but not limiting, τ_b is set as 10 frames, corresponding to about 1 second. To track the stationary object in e.g. a busy site in which the object may be occluded frequently by moving foreground objects, a layer representation based on the object's template image is built. The layer representation of the detected stationary object is defined as

$$L_i = \left\{ A_i^t, T_i, \{d_l^j\}_{j=t-\tau_d}^t, \{d_c^j\}_{j=t-\tau_d}^t, \{d_p^j\}_{j=t-\tau_d}^t, \{s_k\}_{k=t-\tau_s}^t \right\} \quad (1b)$$

where A_t^i is the template image of the object maintained at time t , T_i is the *Principal Colour Representation* (PCR) of the object stored when the object was detected as a stationary object. That is, the template image is based at least on the last frame of the sequence used in detecting the object as a stationary object. d_t^j is the difference measure between the template A_t^j and the frame $I_j(s)$ for the corresponding region of A_t^j , d_c^j is the difference measure between the consecutive frames $I_{j-1}(s)$ and $I_j(s)$ for the region of the template, d_p^j is the visibility measure of the object from the corresponding region in the frame $I_j(s)$, and s_k is an estimated state of the stationary object at time k . Measures in τ_d most current frames and states in τ_s most current frames are recorded. The details of calculation of these measures and estimating states from these measures for each layer object will be described below.

In e.g. a busy public site, if there are some objects staying in the scene, they will often merge with moving objects and the result in a high complexity for object tracking. In separating the layer or stationary objects from moving objects and track the stationary and moving objects separately, the example embodiment can greatly enhance object tracking much. Let $c = I_i(s)$ be the color of a foreground point in the region of i th template image. According to Bayesian rule, the probability of the point belonging to the background is

$$p(b|c) = \frac{p(c|b)p(b)}{p(c)} \quad (2b)$$

where $p(c|b)$ can be obtained from the *Principal Feature Representation* (PFR) of the background. The PFR at each pixel is used to characterize the background. Let $s = (x, y)$ be a pixel of the image. For each type of feature, a table which records the principal feature vectors and their statistics at s

$$T_v(s) = \left\{ p_v^i(b), \{S_v^i(i)\}_{i=1}^{M_v} \right\} \quad (3b)$$

is built, where $p_v^i(b)$ is the learned probability of s belonging to the background ($P_s(b)$) based on the observation of the feature v and $S_v^i(i)$ records the statistics of the M_v most frequent feature vectors at s , Each $S_v^i(i)$ contains three components

$$S_v^t(i) = \begin{cases} p_{v_i}^t = P_s(v_i) \\ p_{v_i|b}^t = P_s(v_i | b) \\ v_i = (v_{i1}, \dots, v_{iD_v}) \end{cases} \quad (4b)$$

where D_v is the dimension of the vector v . The $S_v^t(i)$ in the table are sorted in descending order with respect to the value $p_{v_i}^t$. Hence, the first N_v elements are used as principal features. Three types of features are used in the example embodiment. They are a spectral feature (color), a spatial feature (gradient), and a temporal feature (color co-occurrence), respectively. Among them, color and gradient features are stable for static background parts and color co-occurrence features are suitable for dynamic background parts. Three tables are used to learn the possible principal features of the three types for the background. They are $T_c(s)$, $T_g(s)$, and $T_{cc}(s)$. The color vector is $c = (R_t, G_t, B_t)$ from the input color frame. The gradient vector is $e = (g_x, g_y)$ obtained by Sobel operator. The color co-occurrence vector is $cc = (R_{t-1}, G_{t-1}, B_{t-1}, R_t, G_t, B_t)$ with 32 levels for each color component.

The probability of the pixel becoming a background point at the current frame can be calculated as

$$p(b) = \frac{N_s(b)}{M_s} \quad (5b)$$

where $N_s(b)$ is the background points in a small window W_s centered at s in the previous frame, and M_s is the number of points within the window. Similarly, the probabilities of s belonging to the layer (stationery) object or moving foreground object are

$$p(l | c) = \frac{p(c | l)p(l)}{p(c)} \text{ and } p(f | c) = \frac{p(c | f)p(f)}{p(c)} \quad (6b)$$

respectively. The probabilities $p(c | l)$ and $p(c | f)$ can be calculated with Gaussian kernels. Let c_x^l be the color of point x in the template A_t^{l-1} within the window W_s . Then $p(c | l)$ can be calculated as

$$p(\mathbf{c} | l) = \max_{\mathbf{x} \in W_s} \{k_c(\mathbf{c}_x^l - \mathbf{c})k_s(\mathbf{x} - \mathbf{s})\} \quad (7b)$$

where the Gaussian kernels can be written as $k_g(\mathbf{v}) = \exp\left(-\frac{\mathbf{v}^2}{2\sigma_g^2}\right)$ with $g=c$ or s

indicating the kernel for color or spatial vector, respectively. Again, let $\mathbf{c}_x^f$ be the color of a point $\mathbf{x}$ in the window W_s and in the region of moving foreground objects from the last frame $I_{t-1}(\mathbf{s})$.

The probability $p(\mathbf{c} | f)$ can be calculated as

$$p(\mathbf{c} | f) = \max_{\mathbf{x} \in W_s} \{k_c(\mathbf{c}_x^f - \mathbf{c})k_s(\mathbf{x} - \mathbf{s})\} \quad (8b)$$

The priors can be calculated as

$$p(l) = \frac{N_s(l)}{M_s} \text{ and } p(f) = \frac{N_s(f)}{M_s} \quad (9b)$$

where $N_s(l)$ and $N_s(f)$ are the number of points belonging to the layer object and moving objects within the window W_s in the previous frame.

Comparing Equ (2b) and (6b), it can be found that $p(\mathbf{c})$ has become a common normalization factor. Hence, the likelihoods of $\mathbf{s}$ belonging to background, the layer object, or the moving object can be defined as

$$p'(b | \mathbf{c}) = p(\mathbf{c} | b)p(b), p'(l | \mathbf{c}) = p(\mathbf{c} | l)p(l), \text{ and } p'(f | \mathbf{c}) = p(\mathbf{c} | f)p(f) \quad (10b)$$

respectively. The pixel $\mathbf{s}$ would be assigned according to the greatest likelihood value. The mask for the moving objects is used as the input for moving object tracking.

Stationary objects may also be involved in several changes and interactions with other objects through the sequence. For a non-living object, it may e.g. undergo illumination changes, be occluded and removed by other objects. For a living object, the object may change pose or move bodyparts, or start moving again. During tracking the stationary object, the object's states are estimated and the template image updated correspondently in the example embodiment.

In the example embodiment, five states are used to describe the layer object, they are: *motionless*, *occluded*, *removed*, *inner-motion*, and *start-moving*. The state is estimated according to various change measures from a short sequence of most recent frames.

Let s be a point in template $A_i^t(s)$, of the i th layer object. The difference between the template and a current frame at s can be evaluated as

$$d_i^t(s) = \begin{cases} |A_i^t(s) - I_t(s)| / Th_d, & \text{if } |A_i^t(s) - I_t(s)| < Th_d \\ 1, & \text{otherwise} \end{cases} \quad (11b)$$

where Th_d is the threshold according to image noise. Then, the difference measure between the template and current frame for the layer object is defined as

$$d_i^t = \frac{1}{S_A^i} \sum_{s \in A_i^t} d_i^t(s) \quad (12b)$$

where S_A^i is the size of the template.

Similarly, for a point s in the template $A_i^t(s)$, the difference between consecutive frames at the point is evaluated as

$$d_c^t(s) = \begin{cases} |I_t(s) - I_{t-1}(s)| / Th_d, & \text{if } |I_t(s) - I_{t-1}(s)| < Th_d \\ 1, & \text{otherwise} \end{cases} \quad (13b)$$

The difference measure between consecutive frames for the layer object is defined as

$$d_c^t = \frac{1}{S_A^i} \sum_{s \in A_i^t} d_c^t(s) \quad (14b)$$

The difference measures are calculated on color vectors.

If the changes over the region of the template are caused by motion of the object itself, even if the differences d_i^j and d_c^j would be large, the visibility (visibility measure d_p^j) of the object in the current frame based on PCR would still be high since the PCR is a global representation not related to spatial information. On the other hand, if the changes are caused by occlusion of other objects, the visibility of the layer object in the current frame would be low. Let

T_i be the PCR of the layer object in L_i^t that was stored when the object was detected as a stationary object, and T_i^t be the PCR from the region overlapped by the template A_i^t in the current frame. Then the visibility measure of the layer object in the current can be evaluated as $d_p^t = P(T_i^t | T_i)$. More particular, Let O_m^{t-1} be an object in $I_{t-1}(s)$, and O_n^t be a region in $I_t(s)$. According to Bayesian law, the probability of observing O_m^{t-1} in O_n^t can be computed as

$$P(O_n^t | O_m^{t-1}) = \sum_{i=1}^N P(O_n^t | E_m^i) P(E_m^i | O_m^{t-1}) \quad (15b)$$

From the definition of PCR, the significance of c_m^i for O_m^{t-1} is $P(E_m^i | O_m^{t-1}) = p_m^i / p_m$, and the likelihood of observing O_m^{t-1} in O_n^t according to the evidence of c_m^i is

$$P(O_n^t | E_m^i) = \min\left\{1, p_{c_m^i|n} / p_m^i\right\} = \frac{1}{p_m^i} \min\{p_m^i, p_{c_m^i|n}\} \quad (16b)$$

where $p_{c_m^i|n}$ is the significance of c_m^i from the region O_n^t . Let $C(c_m^i)$ is the subset of the principal colors from O_n^t which match c_m^i . $p_{c_m^i|n}$ can be calculated as

$$p_{c_m^i|n} = \sum_{k: c_n^k \in C(c_m^i)} p_n^k \quad (17b)$$

Now the visibility measure becomes

$$P_{m|n} = P(O_n^t | O_m^{t-1}) = \frac{1}{p_m} \sum_{i=1}^N \min\{p_m^i, p_{c_m^i|n}\} \quad (18b)$$

With the change measures in a short sequence of τ_d most current frames (i.e. image frames from $I_{t-\tau_d}(x)$ to $I_t(x)$) evaluated above, with τ_d normally set to 10 frames in the example embodiment, the states of the tracked layer object are estimated by heuristic rules in the example embodiment:

Rule 1: motionless: If both d_l^j and d_c^j are low through the sequence, it is motionless;

Rule 2: occluded: If both d_l^j and d_c^j turn to moderate or high and d_p^j turns low through the sequence, as well as there are moving objects overlapping the region of the template A_i^t as determined from the bounding boxes of such moving objects in a moving object tracking algorithm applied, the layer object is occluded;

Rule 3: removed: If both d_l^j and d_c^j turn to high and d_p^j turns low, and then d_c^j turns low through the sequence with no moving object overlapping the region of the template, the layer object is removed;

Rule 4: inner-motion: If both d_l^j and d_c^j turn to moderate and then d_c^j turns low through the sequence, while d_p^j keeps being high, this means the layer object has changed its pose or moved part of its body but still stays there;

Rule 5: start moving: If both d_l^j and d_c^j turn and keeps being moderate, and d_p^j keeps being high through the sequence, as well as there is a shift of the layer object, this means the layer object starts moving again.

The parameters for the rules are determined according to a knowledge base of human perceived semantic meanings and an evaluation from real-world videos in the example embodiment. In the example embodiment, but not limiting, for the above rules, the difference measures for d_l^j and d_c^j are low if they are less than 0.25, they are of moderate if they are within (0.25, 0.75), and they are high if they are larger than 0.75. The visibility measure d_p^j is low if it is less than 0.6, otherwise, it is high. The measure of shape shift is calculated by checking the expanding foreground pixels along the boundary of the template A_t^i . If the number of expanded pixels is larger than 50% of the template size, the "shift" of the object is detected. It will be appreciated that for some videos from specific cameras, e.g. cameras with unstable signals, adjustment of the thresholds may be required in different embodiment and as based on the relevant knowledge base.

To track the layer object more robustly in the example embodiment, the layer model is maintained to adapt to real variations of the object without being affected by other objects in the scene. The five most recent states for each layer object ($\tau_s = 5$) are recorded. However, it will be appreciated that other values may be used in different embodiments. If one state has more than 3 supports, the state is confirmed. For the corresponding state, the following updating is performed.

If the layer object is confirmed as being motionless, a smooth operation is performed to the template image. If the object is recognized as being in the inner-motion state, the new image of the object in the current frame will replace the template. If the object is occluded, no updating will be performed. If the object is classified as start-moving, the object will be transformed as a moving object with the same ID and corresponding PCR, mask, and position for tracking by a

moving object tracking algorithm. The layer representation of the object will be deleted. If the object is detected as removed, the object will be transformed as a disappeared object and its layer representation will be destroyed. With these operations, the target object moving around, staying somewhere for a while, and moving again can be tracked continuously and seamlessly by combining the example embodiment with the moving object tracking algorithm described for the example embodiment.

Figure 6 shows a flow chart 600 illustrating a method of object tracking in a video signal according to the example embodiment. At step 602, it is detected that a tracked moving object has become stationary over a sequence of frames. At step 604, a template image of the stationary object is generated based at least one of the frames in the sequence. At step 606, a state of the stationary object is tracked based on a comparison of the template image with a current frame of the video signal.

Event detection:

The structure diagram of an event detection system 700 implementation incorporating the described example embodiment is shown in Figure 7. It contains four fundamental modules, foreground segmentation module 701, moving object tracking module 702, stationary object tracking module 704, and event detection module 706.

The foreground segmentation module 701 performs the background subtraction and learning and includes the method and system for background updating of the example embodiment described above, applied to e.g. the adaptive background subtraction method proposed in [8]. The background model used in the example implementations employs Principal Feature Representation (PFR) at each pixel to characterize background appearance.

The moving objects are tracked with the deterministic 2.5D multi-object tracking algorithm of the described example embodiment in the moving object tracking module 702. As described above, to deal with great variations of target objects in shapes and scales as well as complex occlusions, moving objects are represented by the models of principal color representation which exploits a few most significant colors and their statistics to characterize the appearance of each tracked object. When a tracked object has been detected as stopping moving, a layer representation, or a template, for the object is established and will be tracked by the stationary object tracking module 704 using the method and system of the described example embodiment. At each time step, the states of templates for the objects are estimated with fuzzy reasoning. The template for one object may shift between five states: motionless, interior motion, occluded, starting moving, and removed. When a template for an object is detected as starting to

move, the template for the object will be deleted and the object will be shifted as a moving object and then tracked by the moving object tracking module 702.

In the event detection module 706, semantic models based on Finite State Machines (FSM) are designed to detect suspected scenarios. In the system 700 of the example implementation, four types of unusual events are detected. They are unattended objects, theft, loitering persons, unattended vehicles or unconscious persons.

An "event" is an abstract symbolic concept of what has happened in the scene. It is the semantic level description of the spatio-temporal concatenation of movements and actions of interesting objects in the scene. Event detection in video understanding is a high level procedure which identifies specific events by interpreting the sequences of observed perceptual features from intermediate level processing. It is a step that bridges the numerical level and the symbolic level. The fundamental part of event detection is event modeling. For an event, the model is determined by the task and the different instantiations. There are generally two issues for event modeling. One is to select an appropriate representation model, or formal language, and the other is to derive the descriptors for the interesting events with the model.

In implementations based on the described example embodiment, unusual events are described by the spatio-temporal evolution of object's states, movements, and actions. On a semantic level, each event can be defined as a sequential succession of a few well-defined states. An event could be started at one or more initial states, and then one state can transit to the next state when new conditions are met as the scene evolves in time. When a specific state is reached, the event is declared. State transition may also happen from an intermediate state back to a previous state if some conditions no more hold for the state. The semantic representation can be modelled based on *Finite State Machines* (FSM). The FSM has at least two advantages: (1) it is explicit and natural for semantic description; (2) FSM can readily and flexibly incorporate a variety of context information from intermediate-level processing.

Using *Finite State Machine*, each specific event can be represented by a directed graph $G_i^e = \langle S_i^e, E_i^e \rangle$, where S_i^e is the set of nodes representing the *states* and E_i^e is the set of edges representing the *transitions*. One example of a FSM 800 is described in Figure 8. Any new object is initiated to be state "0" 802 for all the events defined. This is the initial state. The FSM 800 is truly started only when some conditions are met and the active node transits to the next intermediate state, i.e., state "1" 804. There can be more than one intermediate state for the FSM 800 of an event, depending on the complexity of an event. The FSM 800 reaches the final state "End" 406 when all the conditions are met and then the corresponding specific event is triggered.

The FSM 800 is updated at each new frame. The FSM 800 could have the self-loop transition for each state. Although the FSM 800 could remain at the same state, some or all properties of the object may have changed. At least, a time counter is incremented for each frame.

The more complicated an event, the bigger is N , i.e. the number of intermediate states in the FSM 800, and the more is the chance to deliver an unreliable detection result. Therefore, an important task in event modeling is to trim any unnecessary states by careful analysis and to identify the simplest event model.

The input of an FSM is the numerical perceptual features generated by moving and stationary object tracking modules (compare 702 and 704 in Figure 7). The visual cues of each tracked object can include shape, position, motion, and relations with others. The visual cues in the example implementation are:

- Object ID: the identity number of each tracked foreground object;
- Box: bounding box of the tracked object in current frame;
- Size: the area of the object in current frame;
- Status: indicates whether the tracked object is moving around or stationary;
- StayTime: indicates how long the object has stayed in the scene;
- InGroup: indicates whether the object is an isolated one or merged with others;
- Visibility: a measure within $[0,1]$ indicates the degree of occlusion when overlapping with others;
- Motion: a measure within $[0,1]$ indicates the degree of interior motion of a stationary object.

The general processing flow for event detection in the example implementation is shown in Table III.

An advantage of the tracking modules (compare 702, 704 in Figure 7) is the capability to resume tracking of some objects that are lost for a few frames. The two events, UNATTENDED OBJECT and THEFT, are directly concerned with object disappearance in the example implementation. Thus when an active object does not appear in the track records of the current frame, one preferably determines whether it is temporarily lost or whether there is a genuine disappearance. To achieve this, a first-in-first-out (FIFO) stack is built to contain the track records of N frames. $O_{Tracked}$ are the track records of the previous N -th frame and the triggered event is delayed by N frames. As such, in the example implementation it is possible to 'look forward' to check the case of disappearance of an object, with $N = 30$ in the example implementation. With a processing rate of 8 frames/sec or above, this represents a delay of less than 4 seconds. It will be

appreciated that the delay can be balanced against the accuracy of detection in different implementations.

Loitering Detection

Loitering as defined in the example implementation involves one object. It is defined as a person wandering in the observed scene with the duration $t > T_{Loitering}$. The FSM is initialized for each new object. The FSM has one intermediate state: "Stay" which indicates that the tracked person is staying in the scene, whether moving around or stationary. There are two conditions for the transition from state "INIT" to state "Stay":

- The object is classified as human;
- The object moves in the scene (moving around or staying somewhere with frequent interior motion).

In state "Stay", a time counter t is continuously incremented as new frames are coming in. When $t > T_{Loitering}$, the FSM transits from state "Stay" to state "Loiter" and a loitering event is triggered.

Unconscious Person Detection

As defined in the example implementation, this event also involves one object, a person. It is defined as an object becoming complete static with the duration $t > T_{Static}$. The FSM is initialized for each new object. When the tracked object is recognised as a person, the FSM transits to state "M", which indicates a person who is moving around or has significant interior motion. The second intermediate state of the FSM is "S", which indicates a person becoming and staying static, or complete motionless. There are two conditions for the transition from state "M" to state "S":

- The position of the person does not change compared to the previous frame;
- The interior motion of the person $m < T_{IntMotion}$.

In state "S", a time counter t is continuously incremented as new frames are coming in. When $t > T_{Static}$, the FSM transits from state "S" to state "UP", indicating that an unconscious person is detected. Examples of unconscious person include a sleeping or faint person. It will be appreciated that similar conditions can be used to detect e.g. a vehicle staying overtime in a zone for short stopping, in which case the object of interest is changed to vehicle instead of person.

Unattended Object Detection

This event as defined in the example implementation involves two objects. The FSM is initialized for each new object. When the new small object is identified as being separated from another large moving object, and it stays static, a deposited object is detected and the ownership

is established between the two objects. The FSM transits from state "INIT" to state "Station". In the state, the object is associated with its owner. If the owner leaves the scene covered by the camera, the FSM transits from state "Station" to state "UO" and the 'Unattended Object' is declared.

Theft Detection

This event as defined in the example implementation involves three objects. The FSM is initialized for each new object. Similar to the event of unattended object, when the new small object is identified as being separated from another large moving object, and it stays static, a deposited object is detected and the ownership is established between the two objects. The FSM transits from state "INIT" to state "Station". In the state, the object is associated with its owner. However, when the object disappears and this happens with that another object got it and the owner still stays in the scene, the FSM transits from the state "Station" to the state "Theft" and a 'Theft' event is declared, meanwhile, the second person is identified as the potential thief.

The method and system of the example embodiment can be implemented on a computer system 900, schematically shown in Figure 9. It may be implemented as software, such as a computer program being executed within the computer system 900, and instructing the computer system 900 to conduct the method of the example embodiment.

The computer system 900 comprises a computer module 902, input modules such as a keyboard 904 and mouse 906 and a plurality of output devices such as a display 908, and printer 910.

The computer module 902 is connected to a computer network 912 via a suitable transceiver device 914, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN).

The computer module 902 in the example includes a processor 918, a Random Access Memory (RAM) 920 and a Read Only Memory (ROM) 922. The computer module 902 also includes a number of Input/Output (I/O) interfaces, for example I/O interface 924 to the display 908, and I/O interface 926 to the keyboard 904.

The components of the computer module 902 typically communicate via an interconnected bus 928 and in a manner known to the person skilled in the relevant art.

The application program is typically supplied to the user of the computer system 900 encoded on a data storage medium such as a CD-ROM or flash memory carrier and read utilising a corresponding data storage medium drive of a data storage device 930. The

application program is read and controlled in its execution by the processor 918. Intermediate storage of program data maybe accomplished using RAM 920.

It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

References

- [1] D. Lowe. Distinctive image features from scale-invariant key-points. *Int'l J. Computer Vision*, 60(2):91–110, 2004.
- [2] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Statistical modeling of complex background for foreground object detection. *IEEE Trans. Image Processing*, 13(11):1459–1472, 2004.
- [3] L. Li and M. K. H. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Trans. Image Processing*, 11(2):105–112, 2002.
- [4] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.
- [5] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-Based Object Tracking,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [6] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [7] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [8] Liyuan Li, et. al. *IEEE T-IP*, vol. 13, no. 11, pp. 1459–1472, 2004.
- [9] C. Wren, A. Azarbaygani, T. Darrell, and A. Pentland. *Pfinder*: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

CLAIMS

1. A method of multi-object tracking in a video signal; the method comprising the steps of:

receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked;

generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and

for each parent node having two or more child nodes,

a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;

b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;

c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and

iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;

and then

for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image;

for each child node having two or more corresponding objects assigned thereto,

d) sorting the corresponding objects according to estimated depth in said each child node in said second image;

e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;

f) updating the state and the visual content of the located corresponding object based on the second image;

g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and

iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

2. The method as claimed in claim 1, wherein step a) comprises calculating visible portions of the respective corresponding objects in the first image to derive the estimated depths of the respective corresponding objects.

3. The method as claimed in claim 1, wherein step b) comprises assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node based on a posterior probability evaluated by Bayes rule.

4. The method as claimed in claim 3, wherein the a posterior probability evaluated by Bayes rule is based on principle colour representations (PCRs) of the corresponding object and said one child node respectively.

5. The method as claimed in claim 1, wherein step c) comprises removing the visual content of the assigned corresponding object from the visual data associated with said one child node based on PCRs of the assigned corresponding object and said one child node respectively.

6. The method as claimed in claim 1, wherein step d) comprises calculating visible portions of the respective corresponding objects in the second image to derive the estimated depths of the respective corresponding objects.

7. The method as claimed in claim 1, wherein step e) comprises applying the means-shift calculation to locate the corresponding object having the lowest depth in said each child node based on gravity centres of pixels of each principle colour components in a PCR of the corresponding object in the second image.

8. The method as claimed in claim 1, wherein step g) comprises removing the updated visual content of the located corresponding object from the visual data associated with said each child node based on PCRs of the located corresponding object and said each child node respectively.

9. The method as claimed in claim 1, further comprising storing tracking data including the updated status and visual content of each corresponding object for a series of

consecutive frames and detecting an event in the video signal based on the stored tracking data.

10. The method as claimed in claim 1, further comprising the step of:
for each parent node having no child node, deleting the corresponding object.
11. The method as claimed in claim 1, further comprising the step of:
for each parent node having only one child node, assigning all corresponding objects to said one child node.
12. The method as claimed in claim 1, further comprising the step of:
for each child node having no corresponding object assigned thereto, check whether the object is disappeared, and if not, set a new corresponding object to said each child node.
13. The method as claimed in claim 1, further comprising the step of:
for each child node having only one corresponding object assigned thereto, update the state and visual content of said one corresponding object from the visual data associated with said each child node.
14. A multi-object tracking module for multi-object tracking in a video signal;
the module comprising:
 - means for receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked;
 - means for generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and
 - the means for generating, for each parent node having two or more child nodes,
 - a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;
 - b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;

c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and

iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;

and then, for each child node having only one corresponding object assigned thereto, updating a state and the visual content of said one object based on the second image;

for each child node having two or more corresponding objects assigned thereto,

d) sorting the corresponding objects according to estimated depth in said each child node in said second image;

e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;

f) updating the state and the visual content of the located corresponding object based on the second image;

g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and

iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

15. A data storage medium having stored thereon computer code means for instructing a computer system to execute a method of multi-object tracking in a video signal; the method comprising the steps of:

receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked;

generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and

for each parent node having two or more child nodes,

a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image;

b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node;

c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and

iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes;

and then

for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image;

for each child node having two or more corresponding objects assigned thereto,

d) sorting the corresponding objects according to estimated depth in said each child node in said second image;

e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node;

f) updating the state and the visual content of the located corresponding object based on the second image;

g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and

iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

Method And System For Multi-Object Tracking

ABSTRACT

A method and system for multi-object tracking in a video signal. The method comprises the steps of receiving first and second segmented images of two consecutive frames of the video signal respectively, at least one of the first and second segmented images including one or more foreground regions, each foreground region corresponding to one or more objects to be tracked; generating one or more directed acyclic graphs (DAGs) for zero or more parent nodes in the first segmented image and zero to more child nodes in the second segmented images, each DAG including at least one parent or child node; and for each parent node having two or more child nodes, a) sorting the corresponding objects of the foreground region contributing to said each parent node according to estimated depth in said first image; b) assigning the corresponding object having the lowest depth to one of the child nodes of said each parent node; c) removing a visual content of the assigned corresponding object from the visual data associated with said one child node; and iterating steps b) to c) in order of increasing depth of the corresponding objects for assigning all corresponding objects to the two or more child nodes; and then for each child node having only one corresponding object assigned thereto, update a state and the visual content of said one object based on the second image; for each child node having two or more corresponding objects assigned thereto, d) sorting the corresponding objects according to estimated depth in said each child node in said second image; e) applying a means-shift calculation to locate the corresponding object having the lowest depth in said each child node; f) updating the state and the visual content of the located corresponding object based on the second image; g) removing the updated visual content of the located corresponding object from the visual data associated with said each child node; and iterating steps e) to g) in order of increasing depth of the corresponding objects for locating all corresponding objects in a corresponding region of said each child node.

FIG. 5