Tracking people or objects across multiple cameras and maintaining a track within a camera is a challenging task in applications such as video surveillance, traffic monitoring, event detection and recognition. Some of the major challenges while tracking a target are illumination/scale changes and partial occlusion. In this paper, we propose a novel tracking framework using particle filter to efficiently track an object within a camera and a blob-based target association scheme for tracking across cameras. The proposed particle filter tracking algorithm uses a fragment-based approach to model the target and track it by fusing color and gradient features. The proposed solution incorporates coarser level spatial information by fragmenting each particle and is shown to be useful for tracking under partial occlusion. A fast yet robust model update is employed to overcome illumination changes. Experimental results show (i) the robustness of the fragment-based tracking approach with respect to illumination/scale change and partial occlusion and (ii) tracking persons across two cameras.

Keywords: Particle filter; robust tracking; tracking across cameras; feature fusion; fragment-based modeling.
1. Introduction

Visual tracking is one of the crucial tasks in many computer vision applications such as robot navigation, smart rooms, event detection/recognition, surveillance and monitoring. In today’s scenario, due to increasing security concerns, surveillance and monitoring has become essential. Video surveillance of large facilities such as a campus or airport usually involve deployment of a number of cameras which have either non-overlapping or partially overlapping areas of coverage. In such scenarios, tracking of a moving object continuously requires robust tracking of the object in a video stream till the object goes out of view, followed by identification of the cameras in which the object reappears, and subsequent tracking of the object (handover). Despite the efforts made by researchers in developing multi-camera vision system, computer vision algorithms have proven their limits to work in complex environments. These limitations are mainly due to appearance, illumination changes, partial observability inherent in most real world situations. To address above problems, different approaches using different features and algorithms have been investigated. Many interesting solutions have been proposed to overcome the challenges faced by visual tracking, which can be broadly classified into deterministic and stochastic frameworks. Performance evaluation of such systems has yet to evolve.

Deterministic approaches usually reduce to an optimization problem, e.g., minimizing an appropriate cost function. The definition of the cost function is a critical issue. A common choice is a SSD (Sum of Squared Differences) used in many optical flow based approaches. Mean Shift is an alternative deterministic approach to visual tracking, where the cost function is derived from the distance between color histograms of target and candidate models.

Stochastic tracking approaches often reduces to an estimation problem, like estimating the state of the target. Early works used Kalman filter or its variants for visual tracking. However, this restricts the type of model that can be used. Another successful approach in this area is Sequential Monte Carlo algorithms which can model nonlinear/non-gaussian cases. In probabilistic framework of tracking, particle filters have proven to be of enormous success, where the target’s position (or state) is represented by a posterior density over the space of all locations, and this density is represented by a set of random samples (particles) with associated weights. Hotta has also exploited strength of particle filter to weigh the gabor feature based local classifiers. In solution proposed by Hotta, each particle corresponds to the weight of $M$ local classifiers. These weights are changed adaptively and are shown to be robust to partial occlusion and shadows.

In order to combine the merits of both deterministic and stochastic approaches has proposed the scheme, to first produce a smaller number of particles and then shift the samples towards a close local maxima using mean shift. SSD and Mean Shift approaches have been combined with kalman filter by and the approach is shown to overcome their respective disadvantages.
Robust tracking of targets in the presence of illumination/scale changes and partial occlusion is one of the major challenges in surveillance applications. Usually, the approach is to use a model of the target which is robust to lighting and scaling effects, or adaptively update the model to account for changes in appearance. Selecting the right features play a critical role in target modeling. In general, the most desirable property of visual feature is its uniqueness so that the objects can be distinguished in feature space.

Different research groups have used diverse features to accomplish tracking. For example, object shape is used as feature for contour based target model, while color is used as a feature for histogram based models. Other features such as optical flow, texture and combination of features are also used extensively. Perez used color as the main cue and fused it with motion and sound. In order to handle illumination variations, Poon and Fleet used an edge-based model to infer the 3D shape of people from gray-scale sequences. Dawoud et al. used decision fusion algorithm for target tracking.

Further, the problem of occlusion is handled by fragment based target models. Has proposed effective target representation based on multiple color histograms computed on semi-overlapping image areas and has applied fragment based approach in mean shift and particle filter framework to show its robustness to partial occlusion. A model which adapts to slowly changing appearance by giving more weight to stable image structures was proposed by Jepson et al. Also, adaptation techniques based on color models have been proposed to build robust trackers. Lehuger and Shaohua Dorin have proposed adaptive target models to cater for challenging scenarios such as illumination change and partial occlusion. Work by Lehuger proposes a methodology for target update using a dynamic mixture of color models with two components: one fixed and the other rapidly updated. Though this method is suitable for tracking objects undergoing illumination change, performance of the tracker is not guaranteed when the object undergoes partial occlusion for prolonged duration. A simple linear target model update was proposed in order to overcome target appearance changes. The drawback of the linear update is that the updated target model, a linear combination of estimated region and previous model, smooths out the target histogram with time, and may not represent the actual target.

Numerous application of particle filters have been proposed for tracking using multiple cameras as well. Leoputra et al. have proposed a non-overlapping distributed PF-based tracking system using velocity and color. Further, they were able to achieve target association by matching color histograms. Nummiaro et al. used color-based particle filtering based on multiple target models (multiple views of the object) for real-time tracking of an object in a multicamera environment. Ringer proposed modeling and tracking articulated motion from multiple camera views using both extended kalman filter and particle filters.
As mentioned above, objects/people moving in a video sequence have appearance variations including their pose and size, as well as varying lighting conditions and occlusions. All these variations will increase difficulties and challenges in selecting a general feature to describe the object. In order to overcome these limitations, in this paper, we employ a model in which the target is split into fragments, and in each fragment, different image attributes are fused by an adaptive weighting scheme. Further, the model is continuously updated to handle varying scale and illumination. Fragment based target model incorporates the spatial information and it is possible to track even if the object is partially occluded. To build the target model, we use histogram of color and gradient features which, to some extent, is invariant with respect to scale change.

In a particle filter tracking framework, the weights of the particles are updated on observing a new frame. The objective of this work is to design a particle filter-based tracker that is robust to partial occlusion, object appearance and illumination changes. The proposed system determines the reliability of the cues and weigh them accordingly in order to estimate particle weight. Further, the occluded fragments are detected using a similarity measure based on the Bhattacharya coefficient, and they are not considered for computing the weight of the particle. Finally, a robust, yet fast target model update is done by replacing selected target fragments with the currently estimated fragment.

We also extend our tracking approach to a multi-camera scenario, and associate targets across cameras by matching the blobs in feature space. Stored target models based on multiple image cues are compared with the model obtained for every blob in other cameras. Target is associated with the blob having maximum similarity measure based on Bhattacharya coefficient.

The paper is organized as follows. In Sec. 2, we explain in detail the tracking algorithm which involves foreground-background discrimination, definition of the target model based on fragments, and a novel scheme for updating the weight of the particles in the particle filter framework. The extension of the proposed algorithm to track multiple objects across two cameras is described in Sec. 3. Section 4 provides the experimental results of testing the proposed technique on various video sequences, and finally, Sec. 5 concludes the paper.

2. Object Tracking in a Video Stream

This paper presents a novel system for object tracking in particle filter framework, and consists of the following modules: (i) foreground separation, (ii) definition of the target model, (iii) the particle filter-based tracker, and (iv) adaptation of the target model. At the beginning, we propose a foreground object pixel extraction to define the target model based on reliable colors. For the definition of the target model, the target is divided into fragments, and in each fragment, gradient and color histograms are used to characterize the target. Then, target state is estimated using particle filter-based tracker. For robustness of the tracker with respect to
partial occlusion and illumination changes, we propose a novel scheme for assigning weights to the particles. Further, in order to handle object appearance change, we propose an adaptation step to update the target model. Using updated target model, desired targets can be robustly tracked. In the following subsections, details of these modules are described.

2.1. Foreground separation

To include reliable color information into target model, this section will introduce a method for foreground object pixel extraction. It allows efficient separation of the object to be tracked from its background which is crucial for tracking. Then, a mapping which identifies object pixels can be found and used for effective target model update.

Similar to Ref. 24, we use the joint color histograms of the foreground and background to separate them. Let the joint RGB histograms of the target and background regions be denoted as $h_o$ and $h_b$ (see Fig. 1). The log likelihood $L(x_i)$ that a pixel $x_i$ belongs to the object is

$$L(x_i) = \log \left( \frac{\max\{h_o(b(x_i)), \epsilon\}}{\max\{h_b(b(x_i)), \epsilon\}} \right), \quad (1)$$

where $b(x_i)$ is a function that maps the pixel $x_i$ to its histogram bin. A small constant $\epsilon$ is required to avoid numerical instability. The thresholded likelihood $L_t(x_i)$ is

$$L_t(x_i) = \begin{cases} 1 & : L(x_i) > T \\ 0 & : \text{otherwise} \end{cases}, \quad (2)$$

where $T$ is the threshold (set to 0.8 in our work).

Fig. 1. Foreground Separation: (a) Red box indicates foreground; area between red and green boxes is background, (b) isolated foreground.
The mapping $L_t(x_t)$ identifies the target pixels and we use only these pixels for building and selectively updating the target model.

2.2. Target model

After object pixel extraction, this section will design a target model which facilitates tracking in spite of varying illumination/scale changes and partial occlusion. To this end, in the proposed model, we divide the target into fragments (see Fig. 2) and compute multiple independent cues or features for each fragment. The number of fragments is based on the initial size of the target, and is the same throughout the tracking procedure. A model for each fragment is built and the set of these models acts as the target model. In our work, each fragment is modeled using histograms of gradient and color features.

The main motivation for the above approach is that:

(i) The selection of gradient features makes the model robust to illumination to some extent.

(ii) Independent features can be “fused” in order to give more weight to the one which is more reliable.

(iii) Histograms based on color and gradient features are approximately invariant with respect to scale.

(iv) While tracking, if the object is partially occluded, the fragments which are being occluded can be detected and given less preference.

Let $R$ be the region corresponding to the target we want to track, and $M$ be the number of its fragments. The goal of tracking is to estimate the target state vector $\theta = [x, y, \dot{x}, \dot{y}, \dot{a}]$, where $(x, y)$ denotes the center of target region $R$, $(\dot{x}, \dot{y})$ are velocity components along horizontal and vertical directions, and $\dot{a}$ denotes the scale change. Using the particle filter framework, the state vector is estimated at every instant. The idea behind the proposed model is that, for each fragment, we first obtain gradient and color distributions, and in the particle filter algorithm, weight
the particles by fusing these cues. In the next subsection, we describe the procedure to compute the histogram of gradient features.

2.2.1. Gradient features

Gradient feature is very useful to extract target edge information and to certain extent, it is robust to illumination. In order to make the gradient features sensitive to directions, we use the following masks to generate the directional gradient features. These masks correspond to horizontal, vertical, and two diagonal directions.

\[
\begin{align*}
M_{0^\circ} & = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \\
M_{90^\circ} & = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \\
M_{45^\circ} & = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
M_{135^\circ} & = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}
\end{align*}
\]

Then, given a target template with fragments, a four dimensional gradient histogram is computed using the Epanechnikov kernel centered at each of the fragment center. Each dimension is quantized into 5 non-uniform bins such that the bin-width is finer towards 0 and coarser away from 0 (higher gradient). The Epanechnikov kernel is defined as

\[
K(r) = \begin{cases} 
1 - r^2 & : r < 1 \\
0 & : \text{otherwise},
\end{cases}
\]

where \( r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \) is the distance of the pixel from fragment center.

For the \( i \)th fragment, the gradient distribution \( \{G^i_u(y)\}_{u=1}^p \) centered at location \( y \) is calculated as

\[
G^i_u(y) = C_i \sum_{j=1}^{n_i} K \left( \frac{||y - x_j||}{\lambda} \right) \delta[h(x_j) - u],
\]

where \( K \) is Epanechnikov kernel with bandwidth parameter \( \lambda \), \( \delta \) is the Kronecker delta function, \( h(x_j) \) is the histogram function that assigns gradient at the location \( x_j \) to the corresponding bin, \( n_i \) is the number of pixels in the \( i \)th fragment, \( p \) is the total number of bins (5^4), and \( C_i \) is the normalizing constant given by

\[
C_i = \frac{1}{\sum_{j=1}^{n_i} K \left( \frac{||y - x_j||}{\lambda} \right)}
\]

to ensure that \( \sum_u G^i_u = 1 \).

The gradient distribution for \( M \) fragments of \( R \) are denoted as \( G^1, G^2, \ldots, G^M \) and the set of these distributions is one of the features in the target model. Note that the gradient histogram \( (G^i) \) is four dimensional with 5 bins for each dimension.
2.2.2. Color distribution

In addition to the gradient distribution, we also use the color distribution of foreground pixels (as obtained in Sec. 2.1) for target modeling. Color distribution of foreground pixels is determined for every fragment in a similar fashion to that of the gradient distribution, using the kernel $K(r)$ centered at the fragment center. We use joint color histogram in RGB space where, each dimension is divided into 8 bins. For the $i$th fragment, the color distribution \{${P_i}_u(y)$\} at location $y$ is calculated as

$$P_i^u(y) = C_i \sum_{j=1}^{n_i} K\left(\frac{\|y - x_j\|}{\lambda}\right) \delta[h(x_j) - u],$$

where, $n_i$ is the number of foreground pixels in the $i$th fragment, $r$ is the total number of bins (83), and $h(x_j)$ is the histogram function that assigns color at the location $x_j$ to the corresponding bin.

The normalizing factor $C_i$ ensures that $\sum_{u} P_i^u = 1$. Corresponding to each of the $M$ fragments, we obtain color distributions which are denoted as $P_1^1, P_2^2, ..., P_M^M$. The color histogram ($P_i$) is three dimensional with 8 bins for each of the RGB components. The model of the target is the union of gradient and color histograms of all the fragments. In other words, target model of the region $R$ is

$$M^t = \{G_1^t, G_2^t, ..., G_M^t, P_1^t, P_2^t, ..., P_M^t\}.$$

2.3. Particle filter-based enhanced tracker

Particle filter is an inference technique for estimating the unknown state of a dynamical system at every instant from the current and past observations. At instant $k$, let $\theta_k$ and $y_k$ denote the state and the observation of the system, respectively. The two important components characterizing the system are the state transition and observation models which are defined as:

**State transition model:** $\theta_k = F_k(\theta_{k-1}, u_k)$,

**Observation model:** $y_k = H_k(\theta_k, v_k)$,

where $u_k$ is the system noise, $F_k(\cdot)$ characterizes the system dynamics, $v_k$ is the observation noise and $H_k(\cdot)$ models the observation.

In our case, we wish to track the position and size of an object in a video sequence. To this end, the state of the object is described by $\theta = [x, y, \dot{x}, \dot{y}, \dot{a}]$ and video frames are the observations. Given the state transition model and observation model, the problem reduces to computing the posterior distribution $p(\theta_k|y_k)$. Particle filter is a means to approximate the posterior distribution $p(\theta_k|y_k)$, by a set of $N$ weighted particles $S_k = \{(s_k^i, w_k^i)\}_{i=1}^N$ where the $i$th particle is $s_k^i = [x_k^i, y_k^i, \dot{x}_k^i, \dot{y}_k^i, \dot{a}_k^i]$ and its weight $w_k^i$ satisfies $\sum_{i=1}^N w_k^i = 1$. The notation used for
the components of $s_i^k$ can be interpreted with an example: $x_i^k$ denotes the $x$ position of the $i$th particle at instant $k$.

Given $S_{k-1} = \{(s_i^{k-1}, w_i^{k-1})\}_{i=1}^N$ which is properly weighted, we first resample $S_{k-1}$ to get a new set of samples with equal weights $\{(s_i^{k-1}, \frac{1}{N})\}_{i=1}^N$, and then these samples are propagated to $S_k$ through the state transition model. The new weight is updated as $w_i^k \propto p(y_k|s_i^k)$. The complete algorithm is summarized as follows:

**Particle Filter Algorithm**

- Initialize sample set $S_0 = \{(s_i^0, \frac{1}{N})\}_{i=1}^N$.
- For $k = 1, 2, \ldots$
  - Resample $S_{k-1}$ to obtain $\{(s_i^{k-1}, \frac{1}{N})\}_{i=1}^N$
  - For $i = 1, 2, \ldots, N$
    1. Propagate $s_i^{k-1}$ using $s_i^k = A s_i^{k-1} + u_{k-1}$
    2. Update the weights $w_i^k = p(y_k|s_i^k)$
  - Normalize the weight using $w_i^k = \frac{w_i^k}{\sum_{j=1}^N w_j^k}$
- Stop when end of video stream is reached.

In a typical tracking problem, the target to be tracked will be specified in the first frame. The position of the particles is initialized to the center of the specified target region. Similar to the target model, the candidate model is obtained for all the particles in set $S_k$ as described in Sec. 2.2. At instant $k$, the candidate model obtained for the $i$th particle is denoted as $M_i^k = \{G_1^i, G_2^i, \ldots, G_M^i, P_1^i, P_2^i, \ldots, P_M^i\}$.

In order to track the object, for each particle, we compare the candidate model $M_i^k$ and the target model $M^t$ using a novel scheme based on fusing the information from the gradient and color features. Note that the measure of similarity between the candidate and the target model is the weight assigned to the corresponding particle.

2.3.1. **Fusion of cues**

In the proposed fusion scheme, information from the cues is combined in a weighted manner and cues which are more reliable are assigned more weight by using a similarity measure. In our work, we have used the Bhattacharya coefficient as the similarity measure for comparing the gradient and color histograms. For two $m$-bin normalized histograms $a$ and $b$, the Bhattacharya coefficient is defined as

$$\rho(a, b) = \sum_{u=1}^m \sqrt{a_u b_u}, \quad (8)$$

and the Bhattacharya distance between $a$ and $b$ is

$$d(a, b) = \sqrt{1 - \rho(a, b)}. \quad (9)$$
Consider the \( j \)th fragment of the \( i \)th particle (or candidate). Let the Bhattacharya coefficient between the gradient and color histograms of the \( j \)th fragment of the candidate and target be denoted as \( \rho_{ig}(j) \) and \( \rho_{ic}(j) \), respectively. The similarity measure between the \( j \)th fragment of the target and the candidate is defined as

\[
\beta_{ij}^j = \frac{\rho_{ig}(j) w_g + \rho_{ic}(j) w_c}{w_g + w_c},
\]

(10)

where \( w_g \) and \( w_c \) are the weights assigned to the gradient and color features. In order to give more importance to that feature which is more reliable, these weights are based on the Bhattacharya distance. In our work, \( w_g \) is

\[
w_g = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1 - \rho_{ig}(j)}{2\sigma^2}\right)
\]

(11)

with a similar definition for \( w_c \). The proposed simple linear fusion scheme (10) exploits the reliability of both the cues, and can be easily extended for multiple cues.

By comparing the corresponding fragments of the \( i \)th candidate and the target using (10), we obtain a set of similarity measures for all the fragments, which is denoted as

\[
\rho(\text{target}, s_i^k) = \{\beta_1^j, \beta_2^j, \ldots, \beta_M^j\},
\]

(12)

and these measures are combined to compute the particle weight. Note that when the target is partially occluded, the fragments which are not occluded will have a higher match value compared to the other occluded ones. In the next subsection, we use this property for assigning weights to the fragments, and compute the particle weight based on these fragment weights.

### 2.4. Particle weight update and state estimation

Using the set of similarity measures in (12), the fragment weight associated with the \( j \)th fragment of the \( i \)th candidate model is

\[
w_k^j(j) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1 - \beta_j^j}{2\sigma^2}\right).
\]

(13)

We combine the fragment weights obtained in (13) to assign a weight to the corresponding particle. The obvious approach is to compute the average of all fragment weights and use it as the particle weight. The drawback of this approach is that an occlusion affecting even a single fragment may drift or alter the final particle weight adversely. On the other hand, we would like to employ a robust method which could handle outlier contributions from occluded fragments. To this end, we consider only those fragments whose weights are above a certain threshold. In other words, weight
$w_k^i$ of the particle $s_k^i$ is

$$w_k^i = \frac{\sum_{j \in J} w_k^j(j)}{|J|}, \quad J = \{j | w_k^j(j) \geq T_w, j = 1, 2, \ldots, M\},$$

where the threshold $T_w = 0.9 \times \max_{j \in \{1, 2, \ldots, M\}} \{w_k^j(j)\}$ and $|$ denotes cardinality.

The particle filter approximates the posterior distribution $p(\theta_k | y_k)$ by a set of weighted particles $\{(s_k^i, w_k^i)\}_{i=1}^N$. The target state estimate $\hat{\theta}_k$ is the minimum mean square error (MMSE) estimate

$$\hat{\theta}_k = \theta_k^{\text{mmse}} = E[\theta_k | y_k] \approx \sum_{i=1}^N w_k^i s_k^i.$$  

### 2.5. Particle re-sampling \\& propagation

The posterior distribution of the state $p(\theta_k | y_k)$ is approximated by the particles and their respective weights in $S_k = \{\{s_k^i, w_k^i\}\}_{i=1}^N$. If large number of the weights have very low value, the tracking efficiency reduces drastically. In order to avoid this, we resample the set of particles $S_k$. The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights. The resampling step involves generating a new set $S_k'$ by resampling (with replacement) $N$ times from the approximate discrete representation of $p(\theta_k | y_k)$ given by

$$p(\theta_k | y_k) \approx \sum_{i=1}^N w_k^i \delta(\theta_k - s_k^i),$$

so that $p(s_k'^i = s_k^i) = w_k^i$. The resulting sample is in fact an i.i.d. sample from the discrete density, therefore weights of all the sample are now set to $1/N$ to obtain $S_k' = \{(s_k'^j, \frac{1}{N})\}_{j=1}^N$. Each particle in the set $S_k'$ is evolved according to the state transition model to obtain $S_{k+1}$

$$s_{k+1}^i = A s_k'^i + u_k,$$

where $s_k^i$ and $s_{k+1}^i$ are state vectors for $i$th particle at time instants $k$ and $k+1$ respectively, $A$ defines the deterministic component of the model and $u_k$ is a white Gaussian noise.

### 2.6. Target model adaptation

In many real-world scenarios, the object can undergo a change in its appearance and lighting condition, and as a consequence, the model of the target is very different...
from the one in the first frame. Hence, we must adapt the target model as the object appearance varies. However, if the updation is done slowly, targets whose appearance change rapidly cannot be tracked faithfully, whereas if the updation is done very fast, the algorithm may learn the wrong model.

The fragment-based tracking allows us to have a reliable tracking as well as to update the model quickly. In our proposed approach to adapt the target model, we update the target fragment by considering information only from the foreground pixels and ignoring the background. To this end, at every time step, the fragments in which the fraction of foreground pixels is greater than a threshold are updated. This updation rule can be stated as:

\[
p_{j,k} = \begin{cases} 
\hat{q}_{j,k} & : \frac{1}{n_j} \sum_{i=1}^{n_j} L_t(x_i) > T_f \\
 p_{j,k-1} & : \text{otherwise}
\end{cases}
\]

where \(p_{j,k}\) and \(p_{j,k-1}\) denotes \(j\)th fragment of target at time instant \(k\) and \(k-1\), respectively, \(\hat{q}_{j,k}\) denotes \(j\)th fragment of the estimated region, \(n_j\) is the total number of pixels in \(j\)th fragment. In our work, a choice of 80% for the fraction threshold yielded good results. The main advantages of the above target model update are: (i) it takes both the current and previous models into consideration, (ii) only the foreground information is used, and (iii) the robustness of tracker is improved because there is no dependency on any one particular fragment for tracking.

3. Establishing Correspondences Across Cameras

When the target is in view, the proposed particle filter-based tracker uses color and gradient cues to track the target. We extend the algorithm for tracking a single target to track multiple targets across many cameras. One of the major challenges is to find the correspondence between the targets visible in different cameras in spite of appearance changes.

In our experimental setup, we have deployed two different cameras (handycams) with non-overlapping areas of coverage. Using method of background subtraction we isolate foreground (or moving objects) in the video streams. Small and spurious blobs due to noise are removed by thresholding. For each foreground region, the gradient- and color-based model is computed. When a person goes out of view of one camera, the object models in all the video streams are compared with the stored target models using a similarity measure (Bhattacharya coefficient). The correspondence of the foreground region is established by finding the target model for which the similarity measure is maximum and above a set threshold.

4. Results and Discussion

We test the proposed tracking algorithm using real video sequences comprising of objects undergoing partial occlusion, appearance and illumination changes. A particle filter with 150 samples is used for tracking. The results obtained using
proposed algorithm are compared with those obtained using global color histogram based particle filter. In Figs. 3–6, the solid gray and the dashed dark gray color rectangles represent the output of the proposed algorithm and the particle filter with global color histogram (without target update), respectively.

4.1. Scale change due to object motion

Figure 3 shows the tracking performance under scale change when the target moves towards or away from the camera. The tracker is able to accurately track the size of the object by the fusion of gradient and color cues. Note that the tracking algorithm which depends only on the color cue, estimates a smaller window compared to the actual one.

Fig. 3. Scale change due to object motion. Dashed-dark gray rectangle: Only color cue; Solid-gray rectangle: color and gradient cues.
4.2. Camera zoom with partial occlusion

The proposed algorithm is tested on a sequence which exhibits scale change with partial occlusion. Initially, the target shown in Fig. 5 (frame 1) is partitioned into four fragments. For the entire sequence, the weight plot for the four fragments is shown in Fig. 4. In frame 4 of the sequence, when the face undergoes partial occlusion, the weight corresponding to fragment labeled as two and four drops considerably, while fragment one and three have higher weights. In this case, tracking is maintained by the un-occluded fragments (fragments one and three). This result demonstrates how fragments and fusion of cues provide robust estimate of size and location of the object (Solid-gray rectangle).

4.3. Target model adaptation

The efficacy of target model adaptation is demonstrated in Fig. 6. The figure shows result of tracking algorithm when there is a drastic illumination change. The target update step is robust enough to cope with different orientation of the person’s face (Gray rectangle in Fig. 6). Unlike the proposed algorithm, the particle filter without target adaptation fails to track the face in shaded area (Dark-gray rectangle in Fig. 6).

4.4. Tracking across cameras

In this subsection, we illustrate tracking people across two cameras. The presented scenario comprises of two cameras covering distinct regions. Figure 7 shows the first sequence where the target models are defined. Further, the two persons reappear in
Fig. 5. Camera Zoom with Occlusion. Dark gray: Particle Filter with color cue; Gray: Particle Filter with fragmentation and color + gradient cues.

1. camera 2 one after the other and are detected in frames shown in Fig. 8 using background subtraction technique. The proposed algorithm is able to track the objects in the subsequent frames and also successfully locate the corresponding objects in the second video stream (Fig. 9).

5. Conclusions

In this paper, we have proposed a robust particle filter-based approach for tracking objects in a video stream. At the beginning, a foreground color extraction method is presented to extract reliable object pixels. Then, two features including object color and gradient, were used to represent fragmented target template. Due to the foreground pixel extraction, target model is built using more reliable colors.
and excludes background pixels. Moreover, fragments leads to robust tracking in presence of partial occlusion. The designed target model and model update scheme can effectively represent target with different sizes and orientations. Since it can well record different changes in target appearance, targets can be very effectively tracked in the video sequences. The contributions of this paper are summarized as follows:

(1) A novel particle weight update scheme, after fusing color and gradient cues, was proposed to track targets in particle filter framework. This weight update scheme can filter out most impossible candidate samples from contributing to the final state estimate. Therefore, different from other methods, which
Fig. 7. Camera 01.

Frame No. 23  Frame No. 123  Frame No. 223  Frame No. 360

Fig. 7. Camera 01.

Frame No. 520  Frame No. 612

Fig. 8. Detected targets.

Frame No. 530  Frame No. 625

Frame No. 790  Frame No. 1010

Fig. 9. Camera 02.
updates particle weights based on color target model similarity, the proposed method considers occluded target region, scale and color of target for particle weight assignment and efficiently estimates target’s state.

(2) A target model update was proposed to cater for changing appearance and lighting conditions. According to this model update, an effective target model could be obtained from current estimated target region and previous target model.

Experimental results for different scenarios have proved robustness of our proposed method in target tracking. Further, we have illustrated the results of the proposed algorithm for tracking persons across two cameras.

Future work includes investigating target tracking in multiple cameras environment with overlapped regions. Also, the task of detecting cameras, in which target reappears needs to be explored.

References


Chhabi Nigam received her ME degree in System Science & Automation from Indian Institute of Science, Bangalore, India in 2007. She is working as a scientist in Electronics and Radar Development Establishment Bangalore India since 2001. She has worked on video surveillance systems, real time radar displays and radar tracking applications.
R. Venkatesh Babu received his BE degree in electrical engineering from Bharathiar University, Coimbatore, India, in 1993, the ME degree from Madurai Kamaraj University, Madurai, India, in 1995, and the PhD degree in electrical engineering from the Indian Institute of Science, Bangalore, India, in 2003, followed by post-doctoral experiences in Norway, France, and Singapore. Currently he is offering consultancy as a domain expert for few private firms and a government organization. The main focus of his research career has been continuously progressive technologies for imparting intelligence to image/video analysis.

S. Kumar Raja received his B.Tech. degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India in 1997 and ME degree (Signal Processing) from the Department of Electrical Engineering, Indian Institute of Science, Bangalore, India, in 1999. He is currently pursuing PhD at Vista group, IRISA/INRIA, Rennes, France. His research interests include Computer Vision, Wavelets and Neural Networks.

K. R. Ramakrishnan received his BE, ME, and PhD degrees in electrical engineering from the Indian Institute of Science, Bangalore, India in 1974, 1976, and 1983, respectively. He is currently a Professor with the Department of Electrical Engineering, Indian Institute of Science. His research interests include image processing, computer vision, medical imaging, watermarking, and multimedia communication.