Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 109 (2008) 146–162
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views

Omar Javed [a,*], Khurram Shafique [b], Zeeshan Rasheed [a], Mubarak Shah [b]

[a] *Object Video, 11600 Sunrise Valley Dr. Reston, VA 20171, USA*
[b] *University of Central Florida, Orlando, FL 32816, USA*

Received 7 December 2005; accepted 22 January 2007
Available online 27 February 2007

Abstract

Tracking across cameras with non-overlapping views is a challenging problem. Firstly, the observations of an object are often widely separated in time and space when viewed from non-overlapping cameras. Secondly, the appearance of an object in one camera view might be very different from its appearance in another camera view due to the differences in illumination, pose and camera properties. To deal with the first problem, we observe that people or vehicles tend to follow the same paths in most cases, i.e., roads, walkways, corridors etc. The proposed algorithm uses this conformity in the traversed paths to establish correspondence. The algorithm learns this conformity and hence the inter-camera relationships in the form of multivariate probability density of space–time variables (entry and exit locations, velocities, and transition times) using kernel density estimation. To handle the appearance change of an object as it moves from one camera to another, we show that all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace. This subspace is learned by using probabilistic principal component analysis and used for appearance matching. The proposed approach does not require explicit inter-camera calibration, rather the system learns the camera topology and subspace of inter-camera brightness transfer functions during a training phase. Once the training is complete, correspondences are assigned using the maximum likelihood (ML) estimation framework using both location and appearance cues. Experiments with real world videos are reported which validate the proposed approach.

Keywords: Multi-camera appearance models; Non-overlapping cameras; Scene analysis; Multi-camera tracking; Surveillance

1. Introduction

There is a major effort underway in the vision community to develop fully automated surveillance and monitoring systems [3,1]. Such systems have the advantage of providing continuous 24 h active warning capabilities and are especially useful in the areas of law enforcement, national defence, border control and airport security. One important requirement for an automated surveillance system is the ability to determine the location of each object in the environment at each time instant. This problem of estimating the trajectory of an object as the object moves around in a scene is known as *tracking* and it is one of the major topics of research in computer vision. In most cases, it is not possible for a single camera to observe the complete area of interest because sensor resolution is finite, and the structures in the scene limit the visible areas. Therefore, surveillance of wide areas requires a system with the ability to track objects while observing them through multiple cameras. Moreover, it is usually not feasible to completely cover large areas with cameras having overlapping views due to economic and/or computational reasons. Thus, in realistic scenarios, the system should be able to handle multiple cameras with non-overlapping fields of view. Also, it is preferable that the tracking system does not require camera calibration or complete site modeling, since the luxury of fully calibrated cameras or site models is not available in most situations. In this paper,

* Corresponding author.
E-mail address: omar.javed@gmail.com (O. Javed).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2007.01.003

we present an algorithm that caters for all these constraints by employing inter-camera appearance and space–time relationships to track people across non-overlapping field of views.

1.1. An overview of the proposed approach

Our focus is on the problem of multi-camera tracking in a system of non-overlapping cameras. We assume that the single camera tracking problem is solved. The task of a multi-camera tracker is to establish correspondence between the observations across cameras, i.e., given a set of tracks in each camera, we want to find which of these tracks belong to the same object in the real world. We accomplish this by first using the observations of objects, passing through the system of cameras in a training phase, to discover the relationships between the cameras. For example, suppose two cameras A and B are successively arranged alongside a walkway, see Fig. 1. Suppose people moving along one direction of the walkway that are initially observed in camera A are also observed entering camera B after a certain time interval. People can take many paths across A and B. However, due to physical and practical constraints, people will follow some paths more often than others. Thus, the locations of exits and entrances between cameras, direction of movement and the average time taken to reach from A to B can be used as cues to constrain correspondences. We refer to these cues as *space–time* cues and exploit these cues to learn the inter-camera relationships. The inter-camera relationships are learned in the form of a probability density function (pdf) of space time parameters (i.e., the probability of an object entering a certain camera at a certain time given the location, time and velocity of its exit from another camera) from the training data. Instead of imposing assumptions about the form of this pdf, we let the data 'speak for itself' [20] by estimating the pdf using kernel density estimators. A commonly used cue for tracking in a single camera is the appearance of the objects. Appearance of an object can be modelled by its color or brightness histograms, and it is a function of scene illumination, object geometry, object surface material properties (e.g., surface albedo) and the camera parameters. Among all these, only the object surface material properties remain constant as an object moves across cameras. Thus, the color distribution of an object can be fairly different when viewed from two different cameras. One way to match appearances in different cameras is by finding a transformation that maps the appearance of an object in one camera image to its appearance in the other camera image. However, for a given pair of cameras, this transformation is not unique and also depends upon the scene illumination and camera parameters. In this paper, we show that despite depending upon a large number of parameters, all such transformations lie in a low dimensional subspace for a given pair of cameras. The proposed method learns this subspace of mappings for each pair of cameras from the training data by using probabilistic principal component analysis. Thus, given appearances in two different cameras, and the subspace of brightness transfer functions learned during the training phase, we can estimate the probability that the transformation between the appearances lies in the learnt subspace.

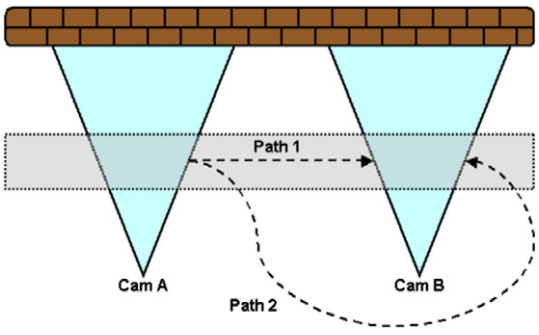

Fig. 1. The figure shows two possible paths an object can take from Camera A to B.

We present an ML estimation framework to use these cues in a principled manner for tracking. The correspondence probability, i.e., the probability that two observations originate from the same object, depends on both the space–time information and the appearance. Tracks assignment is achieved by maximizing the correspondence likelihood. This is achieved by converting the ML estimation problem into a problem of finding the path cover of a directed graph for which an optimal solution can be efficiently obtained.

In Section 2, we discuss related research. In Section 3, a probabilistic formulation of the problem is presented. Learning of inter-camera spatio-temporal and appearance relationships is discussed in Sections 4 and 5, respectively. In Section 6, a maximum likelihood solution to find correspondences is given. Results are presented in Section 7.

2. Related work

In general, multi-camera tracking methods differ from each other on the basis of their assumption of overlapping or non-overlapping views, explicit calibration vs learning the inter-camera relationship, type of calibration, use of 3D position of objects, and/or features used for establishing correspondences. In this paper, we organize the multi-camera tracking literature into two major categories based on the requirement of overlapping or non-overlapping views.

2.1. Multi-camera tracking methods requiring overlapping views

A large amount of work on multi-camera surveillance assumes overlapping views. Jain and Wakimoto [14] used calibrated cameras and an environmental model to obtain 3D location of a person. The fact that multiple views of

the same person are mapped to the same 3D location was used for establishing correspondence. Cai and Aggarwal [2], used multiple calibrated cameras for surveillance. Geometric and intensity features were used to match objects for tracking. These features were modeled as multi-variate Gaussians and the Mahalanobis distance measure was used for matching. Chang and Gong [33] used the top most point on an object detected in one camera to compute its associated epipolar line in other cameras. The distance between the epipolar line and the object detected in the other camera was used to constrain correspondence. In addition, height and color were also used as features for tracking. The correspondences were obtained by combining these features using a Bayesian network. Dockstader and Tekalp [6] also employed Bayesian networks for tracking and occlusion reasoning across calibrated cameras with overlapping views. Sparse motion estimation and appearance were used as features. Mittal and Davis [25] used a region-based stereo algorithm to estimate the depth of points potentially lying on foreground objects and projected them on the ground plane. The objects were located by examining the clusters of the projected points. Kang et al. [17] presented a method for tracking in stationary and pan-tilt-zoom cameras. The ground planes in the moving and stationary cameras were registered. The moving camera sequences were stabilized by using affine transformations. The location of each object was then projected into a global coordinate frame for tracking. The object appearance was modeled by partitioning the object region into its polar representation. In each partition a Gaussian distribution modeled the color variation.

Lee et al. [21] proposed an approach for tracking in cameras with overlapping FOV's that did not require explicit calibration. The camera calibration information was recovered by matching motion trajectories obtained from different views and plane homographies were computed from the most frequent matches. Khan and Shah [19] avoided explicit calibration by using the field of view (FOV) line constraints to handoff labels from one camera to another. The FOV information was learned during a training phase. Using this information, when an object was viewed in one camera, all the other cameras in which the object was visible could be predicted. Tracking in individual cameras was needed to be resolved before handoff could occur.

Most of the above mentioned tracking methods require a large overlap in the FOVs of the cameras. This requirement is usually prohibitive in terms of cost and computational resources for surveillance of wide areas.

2.2. Multi-camera tracking methods for non-overlapping views

To track people in an environment not fully covered by the camera fields of view, Collins et al. [4] developed a system consisting of multiple calibrated cameras and a site model. Normalized cross correlation of detected objects and their location on the 3D site model were used for tracking. Huang and Russel [13] presented a probabilistic approach for tracking vehicles across two cameras on a highway. The solution presented was application specific, i.e., assumption of vehicles travelling in one direction, vehicles being in one of three lanes, and solution formulation for only two calibrated cameras. The appearance was modeled by the mean of the color of the whole object, which is not enough to distinguish between multi-colored objects like people. Transition times were modeled as Gaussian distributions and the initial transition probabilities were assumed to be known. The problem was transformed into a weighted assignment problem for establishing correspondence. Huang and Russel also provided an online version of their correspondence algorithm. The online algorithm trades off correct correspondence accuracy with solution space coverage, which forced them to commit early and possibly make erroneous correspondences. Kettnaker and Zabih [18] used a Bayesian formulation of the problem of reconstructing the paths of objects across multiple cameras. Their system required manual input of the topology of allowable paths of movement and the transition probabilities. The appearances of objects were represented by using histograms. In Kettnaker and Zabih's formulation, the positions, velocities and transition times of objects across cameras were not jointly modeled. However, this assumption does not hold in practice as these features are usually highly correlated.

Ellis et al. [23] determined the topology of a camera network by using a two stage algorithm. First the entry and exit zones of each camera were determined, then the links between these zones across cameras were found using the co-occurrence of entry and exit events. The proposed method assumes that correct correspondences will cluster in the feature space (location and time) while the wrong correspondences will generally be scattered across the feature space. The basic assumption is that if an entry and exit at a certain time interval are more likely than a random chance then they should have a higher likelihood of being linked. Recently, Stauffer [31] proposed an improved linking method which tested the hypothesis that the correlation between exit and entry events that may or may not contain valid object transitions is similar to the expected correlation when there are no valid transitions. This allowed the algorithm (unlike [23]) to handle the case where exit–entrance events may be correlated, but the correlation is not due to valid object transitions. Rahimi and Darrell [28] proposed a method to reconstruct the complete path of an object as it moved in a scene observed by non-overlapping cameras and to recover the ground plane calibration of the cameras. They modeled the dynamics of the moving object as a Markovian process. Given the location and velocity of the object from the multiple cameras, they estimated the trajectory most compatible with the object dynamics using a non-linear minimization scheme. The authors assumed that the objects moved on a ground plane and that all trajectory data of the object was available.

Therefore, the proposed approach was not suitable for an online implementation.

Porikli [27] proposed a method to match object appearances over non-overlapping cameras. In his approach, a brightness transfer function (BTF) is computed for every pair of cameras, such that the BTF maps an observed color value in one camera to the corresponding observation in the other camera. Once such a mapping is known, the correspondence problem is reduced to the matching of transformed histograms or appearance models. However, this mapping, i.e., the BTF varies from frame to frame depending on a large number of parameters that include illumination, scene geometry, exposure time, focal length, and aperture size of each camera. Thus, a single pre-computed BTF cannot usually be used to match objects for moderately long sequences. Recently, Shan et al. [30] presented an unsupervised approach to learn edge measures for appearance matching between non-overlapping views. The matching was performed by computing the probability of two observations from two cameras being generated by the same or different object. Gaussian pdfs were used to compute the same/different probabilities. The proposed solution required the edge images of vehicles to be registered together. Note that the requirement for registering object images might not be possible for non-rigid objects like pedestrians. Moreover, this requirement also constrains the views of the objects in the different cameras to be somewhat similar.

In this paper, we propose inter-camera appearance and space–time relationship models for tracking that do not assume

- explicit camera calibration,
- a site model,
- presence of a single ground plane across cameras,
- a particular non-overlapping camera topology,
- constant illumination, or
- constant camera parameters, for example, focal length or exposure.

In the next section we present a probabilistic formulation of the multi-camera tracking problem.

3. Formulation of the multi-camera tracking problem

Suppose that we have a system of r cameras $C_1, C_2 \ldots, C_r$ with non-overlapping views. Further, assume that there are n objects $p_1, p_2, \ldots, p_n$ in the environment (the number of the objects is not assumed to be known). Each of these objects is viewed from different cameras at different time instants. Assume that the task of single camera tracking is already solved, and let $\mathbf{O}$ be the set of all observations. Moreover, let $O_j = \{O_{j,1}, O_{j,2}, \ldots, O_{j,m_j}\}$ be the set of m_j observations that were observed by the camera C_j. Each observation $O_{j,a}$ is generated by an object in the field of view of camera C_j. The observations consist of two features, appearance of the object $O_{j,a}(app)$ and space–time features of the object $O_{j,a}(st)$ (location, velocity, time etc.). It is reasonable to assume that both $O_{j,a}(app)$ and $O_{j,a}(st)$ are independent of each other. The problem of multi-camera tracking is to find which of the observations in the system of cameras belong to the same object. It is helpful to view the set of observations of each object as a chain of observations with earlier observations preceding the latter ones. The task of grouping the observations of each object can then be seen as linking the consecutive observations in each such chain. Since we have assumed that the single camera tracking problem is solved, the multi-camera tracking task is to link the observations of an object exiting one camera to its observations entering another camera, as the object moves through the system of cameras.

For a formal definition of the problem, let a hypothesized correspondence between two consecutive observations, i.e., exit from one camera and entrance into another, $O_{i,a}$ and $O_{j,b}$, respectively, be denoted as $k_{i,a}^{j,b}$. Moreover, Let $\phi_{k_{i,a}^{j,b}}$ be a binary random variable which is true if and only if $k_{i,a}^{j,b}$ is a valid hypothesis, i.e., $O_{i,a}$ and $O_{j,b}$ are consecutive observations of the same object. We need to find a set of correspondences $K = \{k_{i,a}^{j,b}, \ldots\}$ such that $k_{i,a}^{j,b} \in K$ if and only if $\phi_{k_{i,a}^{j,b}}$ is true.

Let Σ be the solution space of the multi-camera tracking problem. From the above discussion, we know that each observation of an object is preceded or succeeded by a maximum of one observation (of the same object). Hence, if K is a candidate solution in Σ, then for all $\{k_{i,a}^{j,b}, k_{p,c}^{r,e}\} \subseteq K$, $(i,a) \neq (p,c) \wedge (j,b) \neq (r,e)$. In addition, let Φ_K be a random variable which is true if and only if K represents a valid set of correspondences, i.e., all correspondences are correctly established. We want to find a feasible solution in the space Σ of all feasible solutions that maximizes the likelihood, i.e.,

$$K' = \arg\max_{K \in \Sigma} P(\mathbf{O}|\Phi_K = true).$$

Assuming that each correspondence, i.e., a matching between two observations, is conditionally independent of other observations and correspondences, we have:

$$P(\mathbf{O}|\Phi_K = true) = \prod_{k_{i,a}^{j,b} \in K} P\left(O_{i,a}, O_{j,b}|\phi_{k_{i,a}^{j,b}} = true\right). \quad (1)$$

Using the above equation along with the independence of observations $O_{j,a}(app)$ and $O_{j,a}(st)$, for all a and j, we have,

$$P(\mathbf{O}|\Phi_K = true) = \prod_{k_{i,a}^{j,b} \in K} \left(P\left(O_{i,a}(app), O_{j,b}(app)|\phi_{k_{i,a}^{j,b}} = true\right) P\left(O_{i,a}(st), O_{j,b}(st)|\phi_{k_{i,a}^{j,b}} = true\right)\right). \quad (2)$$

Thus the following term gives us the solution:

$$K' = \arg\max_{K \in \Sigma} \prod_{k_{i,a}^{j,b} \in K} \left(P(O_{i,a}(app), O_{j,b}(app)|\phi_{k_{i,a}^{j,b}} = true\right) P\left(O_{i,a}(st), O_{j,b}(st)|\phi_{k_{i,a}^{j,b}} = true\right)$$

This is equivalent to maximizing the following term,

$$K' = \arg\max_{K\in\Sigma} \sum_{k^{j,b}_{i,a}\in K} \log\Big(P(O_{i,a}(app), O_{j,b}(app)|\phi_{k^{j,b}_{i,a}} = true\Big) P\Big(O_{i,a}(st), O_{j,b}(st)|\phi_{k^{j,b}_{i,a}} = true\Big)\Big). \quad (3)$$

In order to obtain the ML estimate we need to know the space–time and appearance probability density functions. This issue is discussed in the next two sections.

4. Learning inter-camera space–time probabilities

Learning is carried out by assuming that the correspondences are known. One way to achieve this is to use only appearance matching for establishing correspondence since space–time relationships between cameras are unknown. Note that during training it is not necessary to correspond all objects across cameras. Only the best matches can be used for learning.

Suppose we have a sample S consisting of n, d dimensional, data points $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n$ from a multi-variate distribution $p(\mathbf{x})$. If the data is continuous, then the Parzen windows technique [7,35] can be used to estimate its density. In our case, the position/time feature vector $\mathbf{x}$, used for learning the space–time pdfs from camera C_i to C_j, i.e., $P(O_{i,a}(sp), O_{j,b}(sp)|\phi_{k^{j,b}_{i,a}} = true)$, is a vector, consisting of the exit and entry locations in cameras, indices of entry and exit cameras, exit velocities, and the time interval between exit and entry events. The camera indices are treated as discrete features while the rest of the vector components are treated as continuous data. Since we have a mixed, i.e., continuous and discrete, data Parzen windows cannot be used directly to estimate the pdf. We have used a mixed density estimator proposed by Li and Racin [22] to obtain the space–time pdf. Let $\mathbf{x} = (\mathbf{x}',\mathbf{x}'')$, where $\mathbf{x}'$ is a d' dimensional vector representing the continuous components of $\mathbf{x}$. $\mathbf{x}''$ is a d'' dimensional vector representing the discrete components and $d = d' + d''$. In addition, Let x''_t be the tth component of $\mathbf{x}''$ and suppose that x''_t can assume $c_t \geqslant 2$ different values, where $(t = 1,2, \ldots,d'')$. The mixed density estimator is defined as

$$\hat{p}(\mathbf{x}) = \frac{1}{n}|\mathbf{H}|^{-\frac{1}{2}} \sum_{i=1}^{n} \kappa(\mathbf{H}^{-\frac{1}{2}}(\mathbf{x}' - \mathbf{x}'_i))\psi(x'', x''_i, \zeta), \quad (4)$$

where the d' variate kernel $\kappa(\mathbf{x}')$, for continuous components, is a bounded function satisfying $\int\kappa(\mathbf{x}')d\mathbf{x}' = 1$, and $\mathbf{H}$ is the symmetric $d'\times d'$ bandwidth matrix. ψ is a multivariate kernel function for discrete components, defined as

$$\psi(\mathbf{x}'', \mathbf{x}''_i, \zeta) = c_0(1-\zeta)^{d''-dif_{i,x}}(\zeta)^{dif_{i,x}} \quad (5)$$

where $dif_{i,x} = d'' - \sum_{t=1}^{d''}\ell(x''_t - x''_{i,t})$, and ℓ is the indicator function, ζ is the scalar discrete bandwidth parameter, and $c_o = \prod_{t=1,\ldots,d''}\frac{1}{c_t-1}$ is a normalization constant.

The multivariate kernel $\kappa(\mathbf{x}')$ can be generated from a product of symmetric univariate kernels κ_u, i.e., $\kappa(\mathbf{x}') = \prod_{j=1}^{d'}\kappa_u(x'_j)$. We use univariate Gaussian kernels to generate $\kappa(\mathbf{x}')$. Moreover, to reduce the complexity, $\mathbf{H}$ is assumed to be a diagonal matrix, i.e., $\mathbf{H} = \text{diag}[h_1^2, h_2^2, \ldots, h_{d'}^2]$, and the smoothing parameter for discrete variables ζ is chosen to be the same for both discrete components. The value of ζ is chosen to extremely small (approaching zero) because we do not want transitions across a pair of cameras being smoothed over and affecting transition probabilities between other cameras.

Each time, a correspondence is made during the training phase, the observed feature is added to the sample S. The observations of an object exiting from one camera and entering into another are separated by a certain time interval. We refer to this interval as *inter-camera travel time*. Following are some key observations that are modeled by the proposed system.

- The dependence of the inter-camera travel time on the magnitude and direction of motion of the object.
- The dependence of the inter-camera travel time interval on the location of exit from one camera and location of entrance in the other.
- The correlation among the locations of exits and entrances in cameras.

Since the correspondences are known in the training phase, the likely time intervals and exit/entrance locations are learned by estimating the pdf. The reason for using the kernel density estimation approach is that, rather than imposing assumptions, the nonparametric technique allows us to directly approximate the d dimensional density describing the joint pdf. It is also guaranteed to converge to any density function with enough training samples [7]. Moreover, it does not impose any restrictions on the shape of the function, neither does it assume independence between the feature set.

5. Estimating change in appearances across cameras

In addition to the space–time information, we want to model the changes in the appearance of an object from one camera to another. The idea here is to learn the change in the color of objects, as they move between the cameras, from the training data and use this as a cue for establishing correspondences. One possible way of doing this was proposed by Porikli [27]. In his approach, a brightness transfer function (BTF) f_{ij} is computed for every pair of cameras C_i and C_j, such that f_{ij} maps an observed brightness value in Camera C_i to the corresponding observation in Camera C_j. Once such a mapping is known, the correspondence problem is reduced to the matching of transformed histograms or appearance models. Note that a necessary condition, for the existence of a one-to-one mapping of brightness values from one camera to another, is that the objects are planar and only have diffuse reflectance. Moreover, this mapping is not unique and it varies from frame to frame depending on a large number of parameters that include illumination, scene geometry, exposure time, focal

length, and aperture size of each camera. Thus, a single pre-computed mapping cannot usually be used to match objects for any moderately long sequence.

In the following subsections, we show that despite a large number of unknown parameters, all BTFs from a given camera to another camera lie in a low dimensional subspace. Moreover, we present a method to learn this subspace from the training data and use this information to determine how likely it is for observations in different cameras to belong to the same object. In other words, given observations $O_{i,a}(app)$ and $O_{j,b}(app)$ from cameras C_i and C_j, respectively, and given all possible brightness transfer functions from camera C_i to camera C_j, we want to estimate the probability that the observations $O_{i,a}(app)$ and $O_{j,b}(app)$ belong to the same object.

5.1. The space of brightness transfer functions

Let $L_i(p,t)$ denote the scene reflectance at a (world) point p of an object that is illuminated by white light, when viewed from camera C_i at time instant t. By the assumption that the objects do not have specular reflectance, we may write $L_i(p,t)$ as a product of (a) material related terms, $M_i(p,t) = M(p)$ (for example, albedo) and (b) illumination/camera geometry and object shape related terms, $G_i(p,t)$, i.e.,

$$L_i(p,t) = M(p)G_i(p,t). \tag{6}$$

The above given model is valid for commonly used Bi-directional Reflectance Distribution Function (BRDF), such as, the Lambertian model and the generalized Lambertian model [26] (see Table 1). By the assumption of planarity, $G_i(p,t) = G_i(q,t) = G_i(t)$, for all points p and q on a given object. Hence, we may write, $L_i(p,t) = M(p)G_i(t)$. The image irradiance $E_i(p,t)$ is proportional to the scene radiance $L_i(p,t)$ [12], and is given as:

$$E_i(p,t) = L_i(p,t)Y_i(t) = M(p)G_i(t)Y_i(t), \tag{7}$$

where $Y_i(t) = \frac{\pi}{4}\left(\frac{d_i(t)}{h_i(t)}\right)^2 \cos^4 \alpha_i(p,t) = \frac{\pi}{4}\left(\frac{d_i(t)}{h_i(t)}\right)^2 c$, is a function of camera parameters at time t. $h_i(t)$ and $d_i(t)$ are the focal length and diameter (aperture) of lens, respectively, and $\alpha_i(p,t)$ is the angle that the principal ray from point p makes with the optical axis. The fall off in sensitivity due to the term $\cos^4 \alpha_i(p,t)$ over an object is considered negligible [12] and may be replaced with a constant c.

Table 1
Commonly used BRDF models that satisfy Eq. (6)

Model	M	G
Lambertian	ρ	$\frac{I}{\pi}\cos\theta_i$
Generalized Lambertian	ρ	$\frac{I}{\pi}\cos\theta_i[1 - \frac{0.5\sigma^2}{\sigma^2+0.33} + \frac{0.15\sigma^2}{\sigma^2+0.09} \times \cos(\phi_i - \phi_r)\sin\alpha\tan\beta]$

The subscripts i and r denote the incident and the reflected directions measured with respect to surface normal. I is the source intensity, ρ is the albedo, σ is the surface roughness, $\alpha = \max(\theta_i,\theta_r)$ and $\beta = \min(\theta_i,\theta_r)$. Note that for generalized Lambertian model to satisfy Eq. (6), we must assume that the surface roughness σ is constant over the plane.

If $X_i(t)$ is the time of exposure, and g_i is the radiometric response function of the camera C_i, then the measured (image) brightness of point p, $B_i(p,t)$, is related to the image irradiance as

$$B_i(p,t) = g_i(E_i(p,t)X_i(t)) = g_i(M(p)G_i(t)Y_i(t)X_i(t)), \tag{8}$$

i.e., the brightness, $B_i(p,t)$, of the image of a world point p at time instant t, is a nonlinear function of the product of its material properties $M(p)$, geometric properties $G_i(t)$, camera parameters, $Y_i(t)$ and $X_i(t)$. Consider two cameras, C_i and C_j, assume that a world point p is viewed by cameras C_i and C_j at time instants t_i and t_j, respectively. Since material properties M of a world point remain constant, we have,

$$M(p) = \frac{g_i^{-1}(B_i(p,t_i))}{G_i(t_i)Y_i(t_i)X_i(t_i)} = \frac{g_j^{-1}\left(B_j(p,t_j)\right)}{G_j(t_j)Y_j(t_j)X_j(t_j)}. \tag{9}$$

Hence, the brightness transfer function from the image of camera C_i at time t_i to the image of camera C_j at time t_j is given by:

$$\begin{aligned} B_j(p,t_j) &= g_j\left(\frac{G_j(t_j)Y_j(t_j)X_j(t_j)}{G_i(t_i)Y_i(t_i)X_i(t_i)} g_i^{-1}(B_i(p,t_i))\right) \\ &= g_j\left(w(t_i,t_j)g_i^{-1}(B_i(p,t_i))\right), \end{aligned} \tag{10}$$

where $w(t_i,t_j)$ is a function of camera parameters and illumination/scene geometry of cameras C_i and C_j at time instants t_i and t_j, respectively. Since Eq. (10) is valid for any point p on the object visible in the two cameras, we may drop the argument p from the notation. Also, since it is implicit in the discussion that the BTF is different for any two pair of frames, we will also drop the arguments t_i and t_j for the sake of simplicity. Let f_{ij} denote a BTF from camera C_i to camera C_j, then,

$$B_j = g_j\left(wg_i^{-1}(B_i)\right) = f_{ij}(B_i). \tag{11}$$

In this paper we use a non-parametric form of the BTF by sampling f_{ij} at a set of fixed increasing brightness values $B_i(1) < B_i(2) < \cdots < B_i(n)$, and representing it as a vector. That is, $(B_j(1),\ldots,B_j(n)) = (f_{ij}(B_i(1)),\ldots,f_{ij}(B_i(n)))$. We denote the space of brightness transfer functions (SBTF) from camera C_i to camera C_j by Γ_{ij}. It is easy to see that the dimension of Γ_{ij} can be at most $d_{\max}$, where $d_{\max}$ is the number of discrete brightness values (For most imaging systems, $d_{\max} = 256$). However, the following theorem shows that BTFs actually lie in a small subspace of the $d_{\max}$ dimensional space (Please see Appendix A for proof).

Theorem 1. *The subspace of brightness transfer functions Γ_{ij} has dimension at most m if for all $a, x \in \mathbb{R}$, $g_j(ax) = \sum_{u=1}^{m} r_u(a)s_u(x)$, where g_j is the radiometric response function of camera C_j, and for all u, $1 \leqslant u \leqslant m$, r_u and s_u are arbitrary but fixed 1D functions.*

From Theorem 1, we see that the upper bound on the dimension of subspace depends on the radiometric response function of camera C_j. Though, the radiometric response functions are usually nonlinear and differ from

one camera to another. They do not have exotic forms and are well-approximated by simple parametric models. Many authors have approximated the radiometric response function of a camera by a gamma function [8,24], i.e., $g(x) = \lambda x^{\gamma} + \mu$. Then, for all $a, x \in \mathbb{R}$,

$$g(ax) = \lambda(ax)^{\gamma} + \mu = \lambda a^{\gamma} x^{\gamma} + \mu = r_1(a)s_1(x) + r_2(a)s_2(x),$$

where, $r_1(a) = a^{\gamma}$, $s_1(x) = \lambda x^{\gamma}$, $r_2(a) = 1$, and $s_2(x) = \mu$. Hence, by Theorem 1, if the radiometric response function of camera C_j is a gamma function, then the SBTF Γ_{ij} has dimensions at most 2. As compared to gamma functions, polynomials are a more general approximation of the radiometric response function. Once again, for a degree q polynomial $g(x) = \sum_{u=0}^{q} \lambda_u x^u$ and for any $a, x \in R$, we can write $g(ax) = \sum_{u=0}^{q} r_u(a)s_u(x)$ by putting $r_u(a) = a^u$ and $s_u(x) = \lambda_u x^u$, for all $0 \leqslant u \leqslant q$. Thus, the dimension of the SBTF Γ_{ij} is bounded by one plus the degree of the polynomial that approximates g_j. It is stated in [10] that most of the real world response functions are sufficiently well approximated by a low degree polynomial, e.g., a polynomial of degree less than or equal to 10. Thus, given our assumptions, the space of inter-camera BTFs will also be a polynomial of degree less than or equal to 10.

In Fig. 2, we show empirically that the assertions made in this subsection remain valid for real world radiometric response functions. In the next subsection, we will give a method for estimating the BTFs and their subspace from training data in a multi-camera tracking scenario.

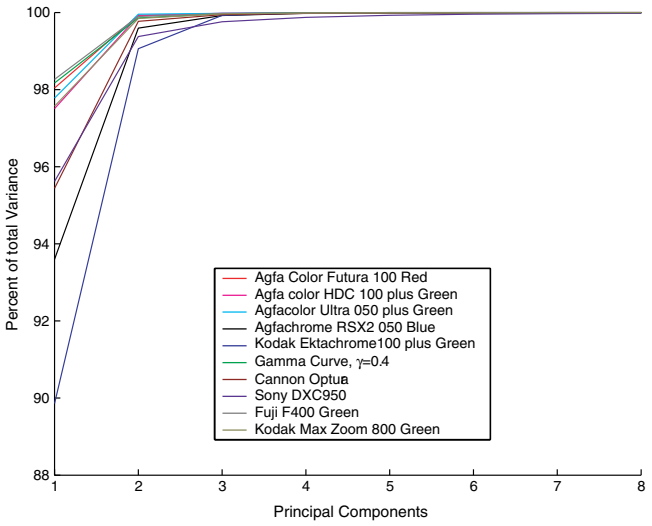

Fig. 2. Plots of the percentage of total variance accounted by m principal components (x-axis) of the subspace of brightness transfer functions from synthetic camera C_1 to camera C_i. Note that each synthetic camera was assigned a radiometric response function of a real world camera/film and a collection of BTFs was generated between pairs of synthetic cameras by varying w in the Eq. (11). PCA was performed on this collection of BTFs. The plot confirms that a very hight percentage of total variance is accounted by first 3 or 4 principal components of the subspace. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

5.2. Estimation of inter-camera BTFs and their subspace

Consider a pair of cameras C_i and C_j. Corresponding observations of an object across this camera pair can be used to compute an inter-camera BTF. One way to determine this BTF is to estimate the pixel to pixel correspondence between the object views in the two cameras (see Eq. (11)). However, self occlusion, change of scale and geometry, and different object poses can make finding pixel to pixel correspondences from views of the same object in two different cameras impossible. Thus, we employ normalized histograms of object brightness values for the BTF computation. Such histograms are relatively robust to changes in object pose [32]. In order to compute the BTF, we assume that the percentage of image points on the observed object $O_{i,a}(app)$ with brightness less than or equal to B_i is equal to the percentage of image points in the observation $O_{j,b}(app)$ with brightness less than or equal to B_j. Note that, a similar strategy was adopted by Grossberg and Nayar [9] to obtain a BTF between images taken from the same camera of the same view but in different illumination conditions. Now, if H_i and H_j are the normalized cumulative histograms of object observations I_i and I_j, respectively, then $H_i(B_i) = H_j(B_j) = H_j(f_{ij}(B_i))$. Therefore, we have

$$f_{ij}(B_i) = H_j^{-1}(H_i(B_i)), \tag{12}$$

where H^{-1} is the inverted cumulative histogram.

As discussed in the previous sub-section, the BTF between two cameras changes with time due to illumination conditions, camera parameters, etc. We use Eq. (12) to estimate the brightness transfer function $\mathbf{f_{ij}}$ for every pair of observations in the training set. Let F_{ij} be the collection of all the brightness transfer functions obtained in this manner, i.e., $\{\mathbf{f_{(ij)n}}\}, n \in \{1, \ldots, N\}$. To learn the subspace of this collection we use the probabilistic Principal Component Analysis PPCA [34]. According to this model a $d_{\max}$ dimensional BTF $\mathbf{f_{ij}}$ can be written as

$$\mathbf{f_{ij}} = \mathbf{W}\mathbf{y} + \overline{\mathbf{f_{ij}}} + \epsilon. \tag{13}$$

Here $\mathbf{y}$ is a normally distributed q dimensional latent (subspace) variable, $q < d_{\max}$, $\mathbf{W}$ is a $d_{\max} \times q$ dimensional projection matrix that relates the subspace variables to the observed BTF, $\overline{\mathbf{f}_{ij}}$ is the mean of the collection of BTFs, and ϵ is isotropic Gaussian noise, i.e., $\epsilon \sim N(0, \sigma^2\mathbf{I})$. Given that $\mathbf{y}$ and ϵ are normally distributed, the distribution of f_{ij} is given as

$$\mathbf{f_{ij}} \sim \mathcal{N}(\overline{\mathbf{f_{ij}}}, \mathbf{Z}), \tag{14}$$

where $\mathbf{Z} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$. Now, as suggested in [34], the projection matrix $\mathbf{W}$ is estimated as

$$\mathbf{W} = \mathbf{U}_q(\mathbf{E}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R}, \tag{15}$$

where the q column vectors in the $d_{\max} \times q$ dimensional $\mathbf{U}_q$ are the eigenvectors of the sample covariance matrix of $\mathbf{F}_{ij}$, $\mathbf{E}_q$ is a $q \times q$ diagonal matrix of corresponding eigenvalues

$\lambda_1 \ldots, \lambda_q$, and $\mathbf{R}$ is an arbitrary orthogonal rotation matrix which can be set to an identity matrix for computational purposes. The value of σ^2, which is the variance of the information 'lost' in the projection, is calculated as

$$\sigma^2 = \frac{1}{d_{\max} - q} \sum_{v=q+1}^{d_{\max}} \lambda_v. \tag{16}$$

Once the values of σ^2 and $\mathbf{W}$ are known, we can compute the probability of a particular BTF belonging to the learned subspace of BTFs by using the distribution in Eq. (14).

Note that till now we have been dealing with only the brightness values of images and computing the brightness transfer functions. To deal with color images we treat each channel, i.e., R, G, and B separately. The transfer function for each color channel is computed exactly as discussed above. The subspace parameters $\mathbf{W}$ and σ^2 are also computed separately for each color channel. Also note that we do not assume the knowledge of any camera parameters and response functions for the computation of these transfer functions and their subspace.

5.3. Computing object color similarity across cameras using the BTF subspace

The observed color of an object can vary widely across multiple non-overlapping cameras due to change in scene illumination or any of the different camera parameters like gain and focal length. Note that, the training phase provides us the subspace of color transfer functions between the cameras, which models how colors of an object can change across the cameras. During the test phase, if the mapping between the colors of two observations is well explained by the learned subspace then it is likely that these observations are generated by the same object. Specifically, for two observations $O_{i,a}$ and $O_{j,b}$ with color transfer functions (whose distribution is given by Eq. (14)) $\mathbf{f}^R_{i,j}, \mathbf{f}^G_{i,j}$ and $\mathbf{f}^B_{i,j}$, we define the probability of the observations belonging to same object as

$$P_{i,j}(O_{i,a}(app), O_{j,b}(app)|k^{j,b}_{i,a}) = \prod_{\mathrm{ch}\in\{\mathrm{R,G,B}\}} \frac{1}{(2\pi)^{\frac{d}{2}}|\mathbf{Z}^{ch}|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}\left(\mathbf{f}^{ch}_{ij}-\overline{\mathbf{f}^{ch}_{ij}}\right)^T (\mathbf{Z}^{ch})^{-1}\left(\mathbf{f}^{ch}_{ij}-\overline{\mathbf{f}^{ch}_{ij}}\right)}, \tag{17}$$

where $\mathbf{Z} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$. The *ch* superscript denotes the color channel for which the value of $\mathbf{Z}$ and $\overline{\mathbf{f}_{ij}}$ were calculated. For each color channel, the values of $\mathbf{W}$ and σ^2 are computed from the training data using Eqs. (15) and (16), respectively.

6. Establishing correspondences

Recall from Section 3, that the problem of multi-camera tracking is to find a set of correspondences K', such that, each observation is preceded or succeeded by a maximum of one observation, and that maximizes the likelihood, i.e.,

$$K' = \arg\max_{K\in\Sigma} \sum_{k^{j,b}_{i,a}\in K} \log\left(P(O_{i,a}(app), O_{j,b}(app)|\phi_{k^{j,b}_{i,a}} = true\right) P\left(O_{i,a}(st), O_{j,b}(st)|\phi_{k^{j,b}_{i,a}} = true\right)$$

The problem of finding the ML solution can be modeled as a graph theoretical problem as follows: We construct a directed graph such that for each observation $O_{i,a}$, there is a corresponding vertex in the directed graph, while each hypothesized correspondence $k^{j,b}_{i,a}$ is modeled by an arc from the vertex of observation $O_{i,a}$ to the vertex of observation $O_{j,b}$. The weight of this arc of the hypothesized correspondence $k^{j,b}_{i,a}$ is computed from the space–time and appearance probability terms, in the summation in Eq. (3). Note that these probabilities are computed using the methods described in Sections 4 and 5. With the constraint that an observation can correspond to at most one succeeding and one preceding observation, it is easy to see that each candidate solution is a set of directed paths (of length 0 or more) in this graph. Also, since each observation corresponds to a single object, each vertex of the graph must be in exactly one path of the solution. Hence, each candidate solution in the solution space is a set of directed paths in the constructed graph, such that each vertex of the graph is in exactly one path of this set. Such a set is called vertex disjoint path cover of a directed graph. The weight of a path cover is defined by the sum of all the weights of the edges in the path cover. Hence, a path cover with the maximum weight corresponds to the solution of the ML problem as defined in Eq. (3).

The problem of finding a maximum weight path cover can be optimally solved in polynomial time if the directed graph is acyclic [29]. Recall that $k^{j,b}_{i,a}$ defines the hypothesis that the observations $O_{i,a}$ and $O_{j,b}$ are consecutive observations of the same object in the environment, with the observation $O_{i,a}$ preceding the observation $O_{j,b}$. Thus, by the construction of graph, all the arcs are in the direction of increasing time, and hence, the graph is acyclic. The maximum weight path cover of an acyclic directed graph can be found by reducing the problem to finding the maximum matching of an undirected bipartite graph. This bipartite graph is obtained by splitting every vertex v of the directed graph into two vertices v^- and v^+ such that each arc coming into the vertex v is substituted by an edge incident to the vertex v^-, while the vertex v^+ is connected to an edge for every arc going out of the vertex v in the directed graph (The bipartite graph obtained from the directed graph is shown in Fig. 3). The edges in the maximum matching of the constructed bipartite graph correspond to the arcs in the maximum weight path cover of the original directed graph. The maximum matching of a bipartite graph can be found by an $O(n^{2.5})$ algorithm by Hopcroft and Karp [11], where n is the total number of vertices in graph G, i.e., the total number of observations in the system. The method described above, assumes that the entire set of observations is available and hence cannot be used in real time applications. One approach to handle this type of

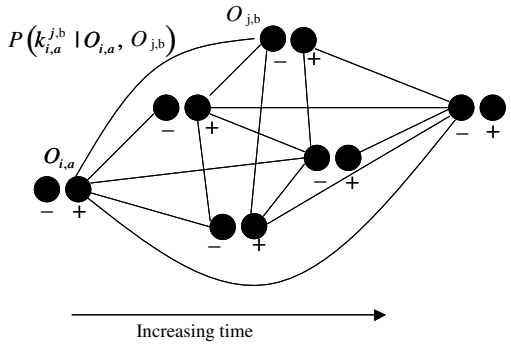

Fig. 3. An example of split graph (constructed from the directed graph) that formulates the multi-camera tracking problem. Each vertex of the directed graph is splitted into + (exit) and − (entry) vertices, such that the + vertex is adjacent to an edge for each arc going out of the vertex and the − vertex is adjacent to an edge for each arc coming into the vertex. The weight of an edge is the same as the weight of the corresponding arc. The graph is bipartite, since no + vertex is adjacent to a + vertex and no − vertex is adjacent to a − vertex.

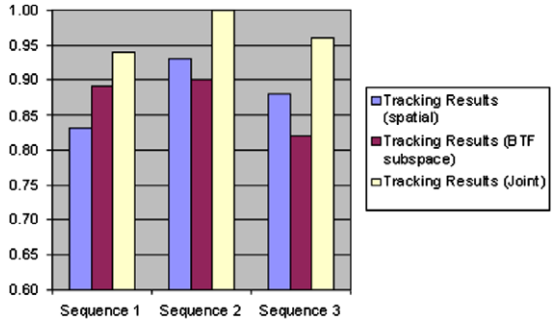

Fig. 4. Tracking results: Tracking accuracy for each of the three sequences computed for three different cases. (1) By using only space–time model, (2) by using only appearance model, and (3) both models. The results improve greatly when both the space–time and appearance models are employed for establishing correspondence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

problem in real time applications is to use a sliding window of a fixed time interval. This approach, however, involves a tradeoff between the quality of results and the timely availability of the output. In order to avoid making erroneous correspondences, we adaptively select the size of sliding window in the online version of our algorithm. This is achieved by examining the space–time pdfs for all observations (tracks) in the environment that are not currently visible in any of the cameras in the system and finding the time interval after which the probability of reappearance of all these observations in any camera is nearly zero. The size of sliding window is taken to be the size of this time interval, and the correspondences are established by selecting the maximum weight path cover of the graph within the window.

7. Results

In this section, we present the results of the proposed method in three different multi-camera scenarios. The scenarios differ from each other both in terms of camera topologies and scene illumination conditions, and include both indoor and outdoor settings. Each experiment consists of a supervised training phase and a testing phase. In both phases, the single camera object detection and tracking information is obtained by using the method proposed in [15]. In the training phase, the known correspondence information is used to compute the kernel density of the space–time features (entry and exit locations, exit velocity and inter-camera time interval) and the subspaces of transfer functions for each color channel (red, blue, and green). In the testing phase, these correspondences are computed using the proposed multi-camera correspondence algorithm. The performance of the algorithm is analyzed by comparing the resulting tracks to the ground truth. We say that an object in the scene is tracked *correctly* if it is assigned a single unique label for the complete duration of its presence in the area of interest. The *tracking accuracy* is defined as the ratio of the number of objects tracked correctly to the total number of objects that passed through the scene.

In order to determine the relative significance of each model and to show the importance of combining the space–time information with the appearance matching scheme, for each multi-camera scenario, the correspondences in the testing phase are computed for three different cases separately, by using (i) only space–time model, (ii) only appearance model, and (iii) both models. The results of each of these cases are analyzed by using the above defined tracking accuracy as the evaluation measure. These results are summarized in Fig. 4 and are explained below for each of the experimental setup.

The first experiment was conducted with two cameras, Camera 1 and Camera 2, in an outdoor setting. The camera topology is shown in Fig. 5(a). The scene viewed by Camera 1 is a covered area under shade, whereas Camera 2 views an open area illuminated by the sunlight (please see Fig. 7). It can be seen from the figure that there is a significant difference between the global illumination of the two scenes, and matching the appearances is considerably difficult without accurate modeling of the changes in appearance across the cameras. Training was performed by using a 5 min sequence. The marginal of the space–time density for exit velocities from Camera 2 and the inter-camera travel time interval is shown in Fig. 5(b). The marginal density shows a strong anti-correlation between the two space–time features and complies with the intuitive notion that for higher velocities there is a greater probability that the time interval will be less, whereas a longer time interval is likely for slower objects. In Fig. 6 the transfer functions obtained from the first five correspondences from Camera 1 to Camera 2 are shown. Note that lower color values from Camera 1 are being mapped to higher color values in Camera 2 indicating that the same object is appearing much brighter in Camera 2 as compared to Camera 1.

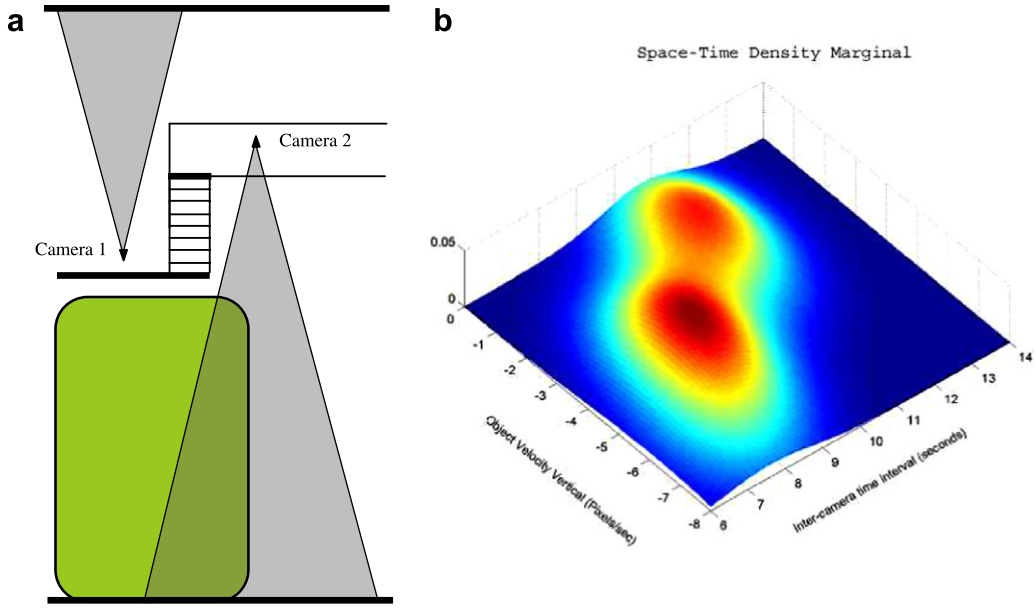

Fig. 5. (a) Two camera configuration for the first experiment. Field-of-view of each camera is shown with triangles. The cameras were mounted approximately 10 yards apart. It took 7–12 seconds for a person walking at normal speed to exit from the view of Camera 1 and enter Camera 2. The green region is the area covered by grass, most people avoid walking over it. (b) The marginal of the inter-camera space–time density (learned from the training data) for exit velocities of objects from Camera 2 and the time taken by the objects to move from Camera 2 to Camera 1. Note if the object velocity is high a lesser inter-camera travel time is more likely, while for objects moving with lower velocities a longer inter-camera travel time is more likely. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

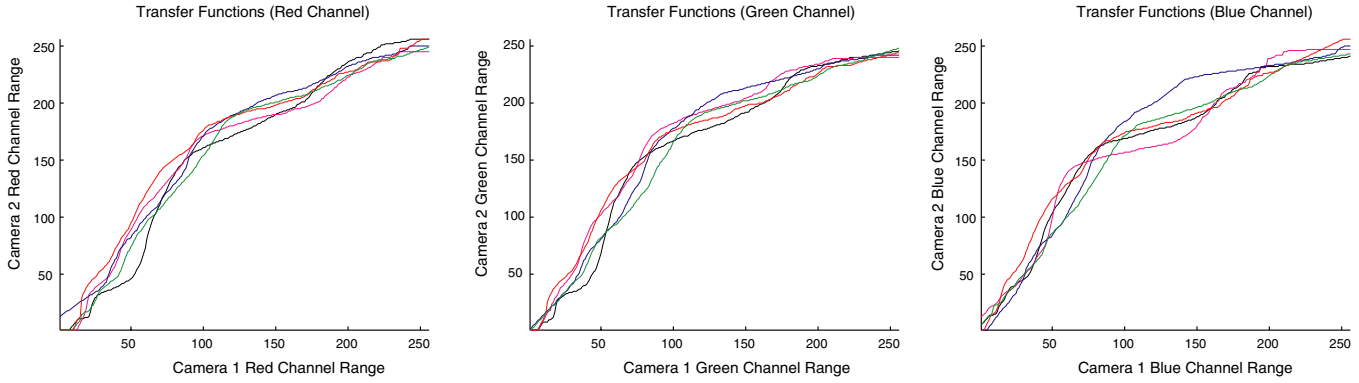

Fig. 6. The transfer functions for the R,G and B color channels from Camera 1 to Camera 2, obtained from the first five correspondences from the training data. Note that mostly lower color values from Camera 1 are being mapped to higher color values in Camera 2 indicating that the same object is appearing much brighter in Camera 2 as compared to Camera 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

The test phase consisted of a 12 minutes long sequence. In this phase, a total of 68 tracks were recorded in the individual cameras and the algorithm detected 32 transitions across the cameras. Tracking accuracy for the test phase is shown in Fig. 4.

Our second experimental setup consists of three cameras, Camera 1, Camera 2, and Camera 3, as shown in Fig. 8(a). The field-of-view of each camera is also shown in the figure. It should be noted that there are several paths from one camera to the other, which make the sequence more complex. Training was done on a 10 min sequence in the presence of multiple persons. Fig. 8(b) shows the probabilities of entering Camera 2 from Camera 1, that were obtained during the training phase. Note that people like to take the shortest possible path between two points. This fact is clearly demonstrated by the space–time pdf, which shows a correlation between the y-coordinates of the entry and exit locations of the two cameras. That is, if an object exits Camera 1 from point A, it is more probable that it will enter Camera 2 at point C rather than point D. The situation is reversed if the object exits Camera 1 from point B. Testing was carried out on a 15 min sequence. A total of 71 tracks in individual cameras were obtained and the algorithm detected 45 transitions within the cameras. The trajectories of the people moving through the scene in the testing phase are shown in Fig. 9. Note that

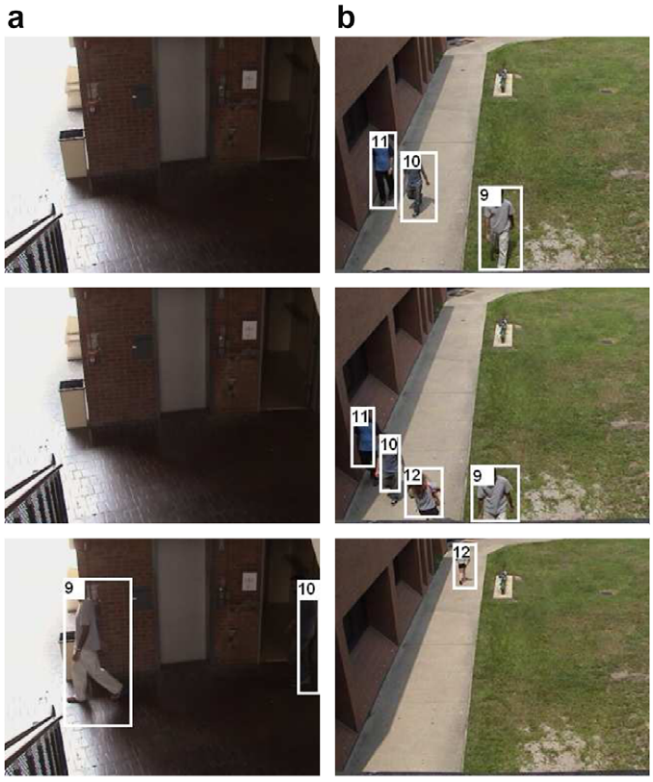

Fig. 7. Frames from sequence 1. Note that multiple persons are simultaneously exiting from Camera 2 and entering at irregular intervals in Camera 1. The first camera is overlooking a covered area while the second camera view is under direct sun light, therefore the observed color of objects is fairly different in the two views (also see Fig. 12). Correct labels are assigned in this case due to accurate color modeling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

people did not stick to a narrow path between Camera 1 and Camera 2, but this did not affect the tracking accuracy and all the correspondences were established correctly when both space–time and appearance models were used (see Fig. 4). Fig. 15 shows some tracking instances in this sequence. In the third experiment, three cameras Camera 1, Camera 2, and Camera 3 were used for an indoor/outdoor setup. Camera 1 was placed indoor while the other two cameras were placed outdoor. The placements of the cameras along with their fields of view are shown in Fig. 10. Training was done on an 8 min sequence in the presence of multiple persons. Testing was carried out on a 15 min sequence. Fig. 11 shows some tracking instances for the test sequence. The algorithm detected 49 transitions among the total of 99 individual tracks that were obtained during this sequence, out of which only two correspondences were incorrect. One such error was caused by a person staying, for a much longer than expected duration, in an unobserved region. That is, the person stood in an unobserved region for a long time and then entered another camera but the time constraint (due to the space–time model) forced the assignment of a new label to the person. Such a scenario could have been handled if there were similar examples in the training phase. The aggregate tracking results for the sequence are given in Fig. 4. It is clear from Fig. 4 that both the appearance and space–time models are important sources of information as the tracking results improve significantly when both the models are used jointly.

In Table 2, we show the number of principal components that account for 99% of the total variance in the inter-camera BTFs computed during the training phase.

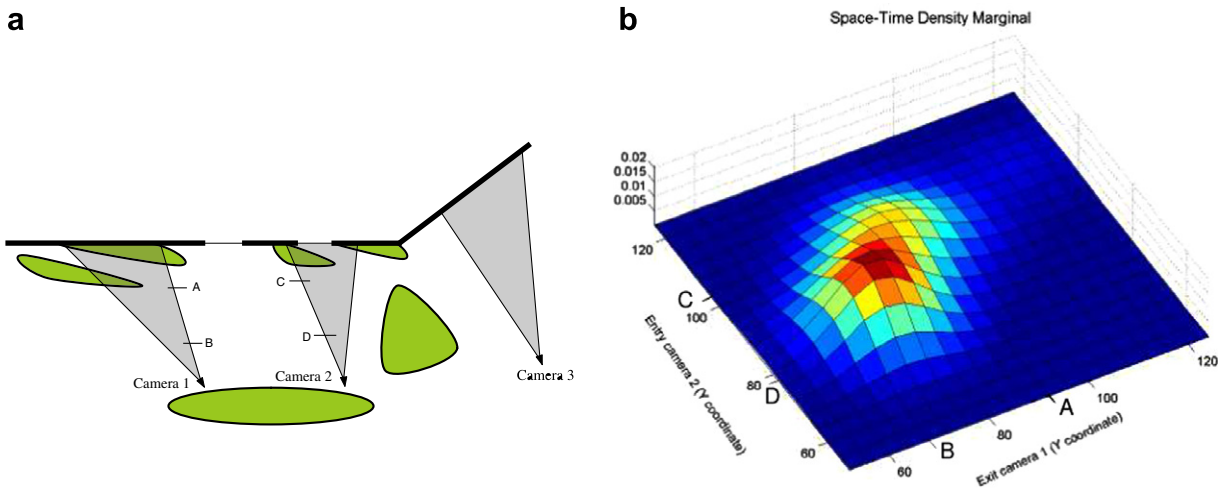

Fig. 8. (a) Camera setup for sequence 2. Camera 2 and Camera 3 were mounted approximately 30 yards apart, while the distance between Camera 1 and Camera 2 was approximately 20 yards. It took 8–14 seconds for a person walking at normal speed to exit from the view of Camera 1 and enter Camera 2. The walking time between Camera 2 and 3 was between 10 and 18 s. The green regions are patches of grass. The points A and D are locations where people exited and/or entered the camera field of view. (b) The marginal of the inter-camera space–time density for exit location of objects from Camera 1 and Entry location in Camera 2. In the graph the y coordinates of right boundary of Camera 1 and left boundary of Camera 2 are plotted. Since most people walked in a straight path from Camera 1 to Camera 2, i.e., from locations A to C and from B to D as shown in (a), thus corresponding locations had a higher probability of being the exit/entry locations of the same person. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Fig. 9. Trajectories of people for the camera setup 2. Trajectories of the same person are shown in the same color. There were a total of 27 people who walked through the environment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

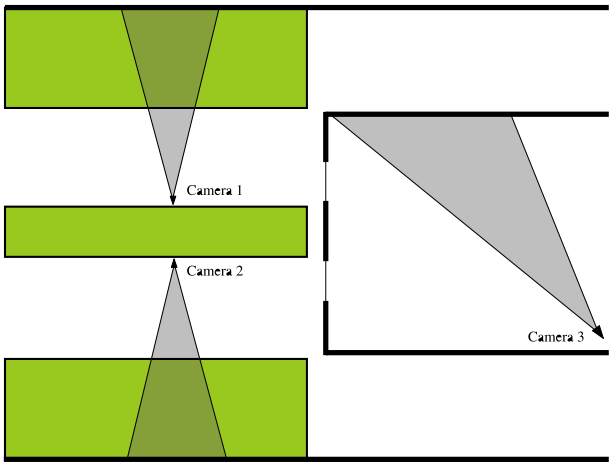

Fig. 10. Camera setup for sequence 3. It is an *Indoor/Outdoor Sequence*. Camera 3 is placed indoor while Cameras 1 and 2 are outdoor. The distance between camera 3 and the other two cameras is around 20 m.

About 40 correspondences were used for training, for each camera pair between which there was a direct movement of people, i.e., without going through an intermediate camera view. Even though the experimental setup does not follow the assumptions of Section 5, such as planarity of objects, the small number of principal components indicates that the inter-camera BTFs lie in a low dimension subspace even in more general conditions.

In order to demonstrate the superiority of the subspace based method we compare it with the direct use of colors for tracking. For direct color base matching, instead of using Eq. (17) for the computation of appearance probabilities $P_{i,j}(O_{i,a}(app), O_{j,b}(app)|k_{i,a}^{j,b})$, we define it in terms of the Bhattacharraya distance between the normalized histograms of the observations $O_{i,a}$ and $O_{i,b}$, i.e.,

$$P_{i,j}(O_{i,a}(app), O_{j,b}(app)|k_{i,a}^{j,b}) = \gamma e^{-\gamma D(h_i,h_j)}, \tag{18}$$

where h_i and h_j are the normalized histograms of the observations $O_{i,a}$ and $O_{j,b}$ and D is the modified Bhattacharraya distance [5] between two histograms and is given as

$$D(h_i, h_j) = \sqrt{1 - \sum_{v=1}^{m} \sqrt{\hat{h}_{i,v}\hat{h}_{j,v}}}, \tag{19}$$

where m is the total number of bins. The Bhattacharraya coefficient ranges between zero and one and is a metric.

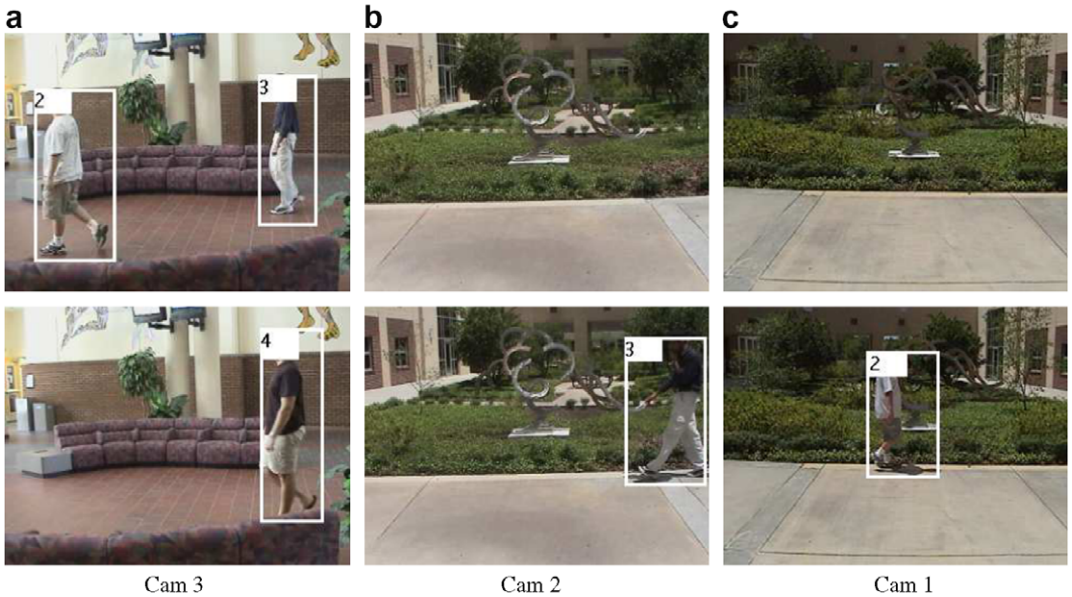

Fig. 11. Frames from sequence 3 test phase. A person is assigned a unique label as it moves through the camera views. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Table 2
The number of principal components that account for 99% of the variance in the BTFs

Sequence No.	Camera pair	No. of principal components (Red)	No. of principal components (Green)	No. of principal components (Blue)
1	1-2	6	5	5
2	1-2	7	7	7
2	2-3	7	7	6
3	1-3	7	6	7
3	2-3	7	7	7

Note that for all camera pairs a maximum of 7 principal components were sufficient to account for the subspace of the BTFs.

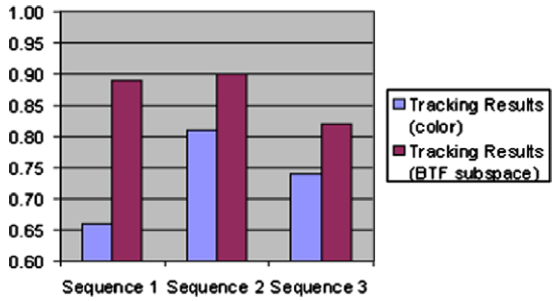

Fig. 13. Tracking accuracy: comparison of the BTF subspace based tracking method to simple color matching. A much improved matching is achieved in the transformed color space relative to direct color comparison of objects. The improvement is greater in the first sequence due to the large difference in the scene illumination in the two camera views. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Once again, the tracking accuracy was computed for all three multi-camera scenarios using the color histogram based model (Eq. (18)). The comparison of the proposed appearance modeling approach with the direct color based appearance matching is presented in Fig. 13, and clearly shows that the subspace based appearance model performs significantly better.

For further comparison of the two methods, we consider two observations, O_a and O_b, in the testing phase, with histograms $H(O_a)$ and $H(O_b)$, respectively. We first compute a BTF, $\mathbf{f}$, between the two observations and reconstruct the BTF, $\mathbf{f}^*$, from the subspace estimated from the training data, i.e., $\mathbf{f}^* = \mathbf{W}\mathbf{W}^T(\mathbf{f} - \bar{\mathbf{f}}) + \bar{\mathbf{f}}$. Here $\mathbf{W}$ is the projection matrix obtained in the training phase. The first observation O_a is then transformed using $\mathbf{f}^*$, and the histogram of the object O_b is matched with the histograms of both O_a and $\mathbf{f}^*(O_a)$ by using the Bhattacharraya distance. When both the observations O_a and O_b belong to the same object, the transformed histogram gives a much better match as compared to direct histogram matching, as shown in Figs. 12 and 14. However, if the observations O_a and O_b belong to different objects then the BTF is reconstructed poorly, (since it does not lie in the subspace of valid BTFs), and the Bhattacharraya distance for the transformed observation either increases or does not change significantly. The

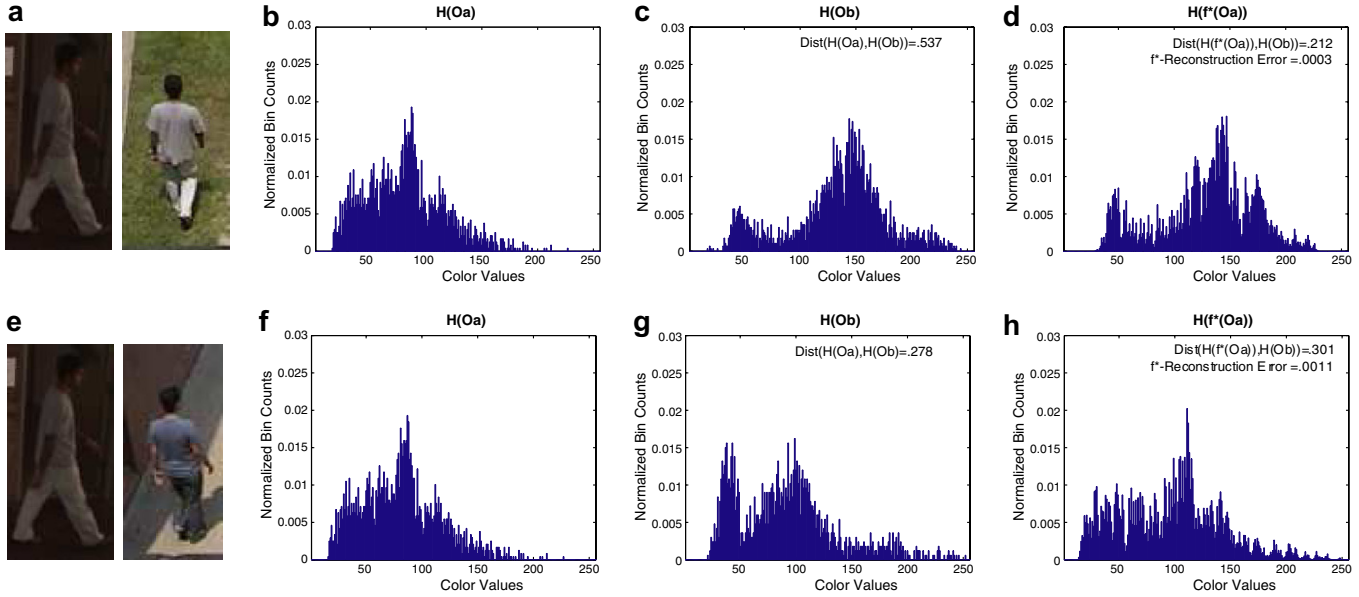

Fig. 12. (a) Observations O_a and O_b of the same object from Camera 1 and Camera 2, respectively from camera setup 1. (b) Histogram of observation O_a (All histograms are of the Red color channel). (c) Histogram of observation O_b. The Bhattacharraya distance between the two histograms of the same object is 0.537. (d) The Histogram of O_a after undergoing color transformation using the BTF reconstruction from the learned subspace. Note that after the transformation the histogram of $(f^*(O_a))$ looks fairly similar to the histogram of O_b. The Bhattacharraya distance reduces to 0.212 after the transformation. (e) Observation from Camera 1 matched to an observation from a different object in camera 2. (f and g) Histograms of the observations. The distance between histograms of two different objects is 0.278. Note that this is less than the distance between histograms of the same object. (h) Histogram after transforming the colors using the BTF reconstructed from the subspace. The Bhattacharraya distance increases to 0.301. Simple color matching gives a better match for the wrong correspondence. However, in the transformed space the correct correspondence gives the least bhattacharraya distance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

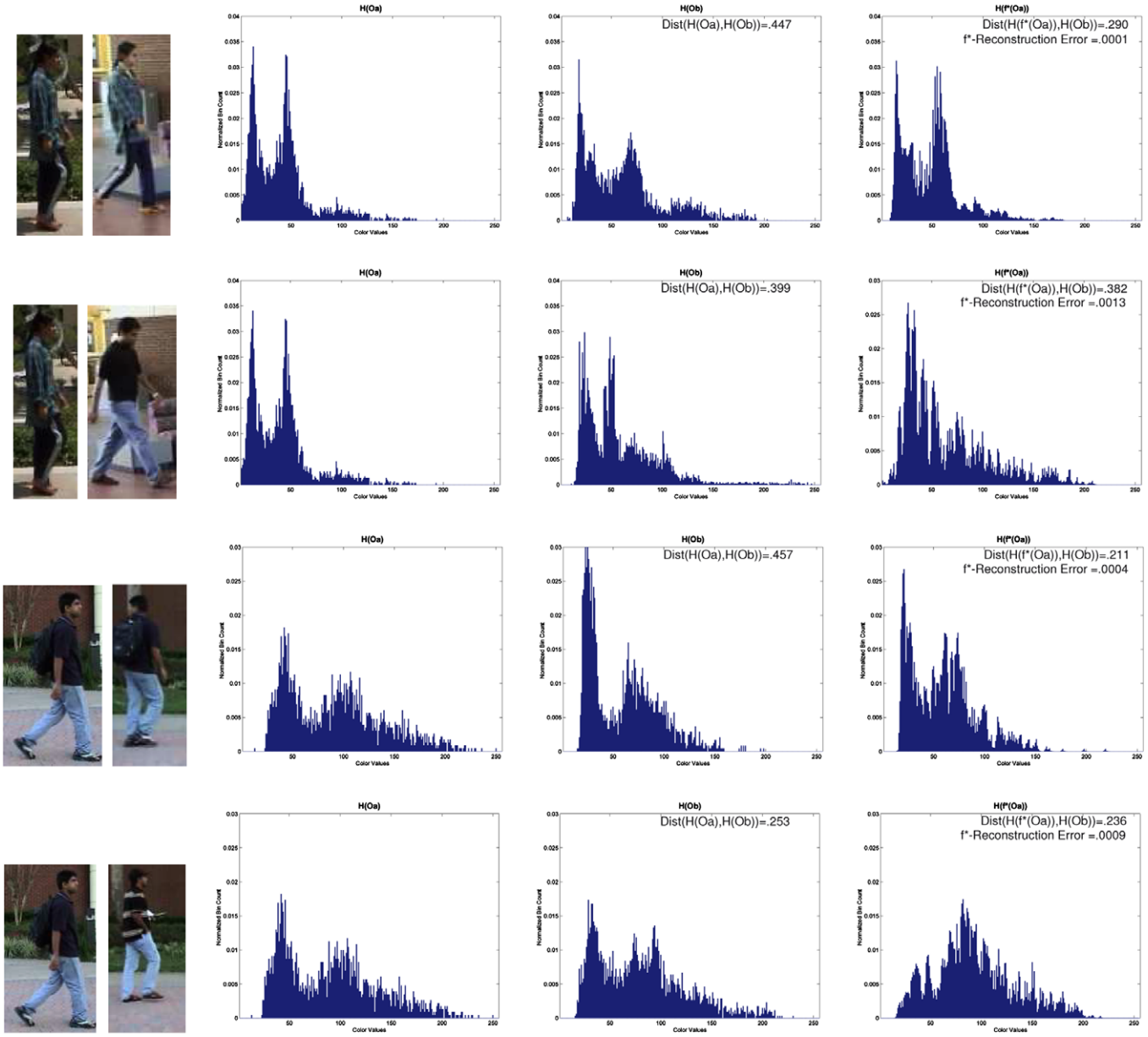

Fig. 14. Row 1: Observations from camera setup 3. The observations are of the same object from Camera 1 and Camera 2, respectively. Their blue channel histograms are also shown. The last histogram is obtained after transforming the colors with a reconstructed BTF f^* from the subspace. Note that the Bhattacharraya distance (shown at the top of the histograms) improves significantly after the transformation. Row 2: Observations of different objects and their blue channel histograms. Here there is no significant change in the Bhattacharraya distance after the transformation. Rows (3,4): Observations from camera setup 2. Here the direct use of color histograms results in a better match for the wrong correspondence. However after the color transformation, histograms of the same objects have the lesser Bhattacharraya distance between them. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

normalized reconstruction error of the BTF, $\mathbf{f}^*$-Reconstruction Error $= \|\mathbf{f} - \mathbf{f}^*\|/\tau$, where τ is a normalizing constant, is also shown in the figures. The aggregate results for the reconstruction error, for the BTFs between the same object and also between different objects are given in Table 3. The above discussion suggests the applicability of the BTF subspace for the improvement of any multi-camera appearance matching scheme that uses color as one of its components.

Our multi-camera tracking system uses a client–server architecture, in which a client processor is associated with each camera. The advantage of this architecture is that the computationally expensive tasks of object detection and single camera tracking are performed at the client side, while the server only performs the multi-view correspondence. The communication between the client and server consists of histogram and trajectory information of objects, and this information is sent only when the objects exit or enter the field of view of a camera. Note that, the server does not use the images directly and thus the communication overhead is low. In our experiments, there was no significant difference in frame rates between the two camera

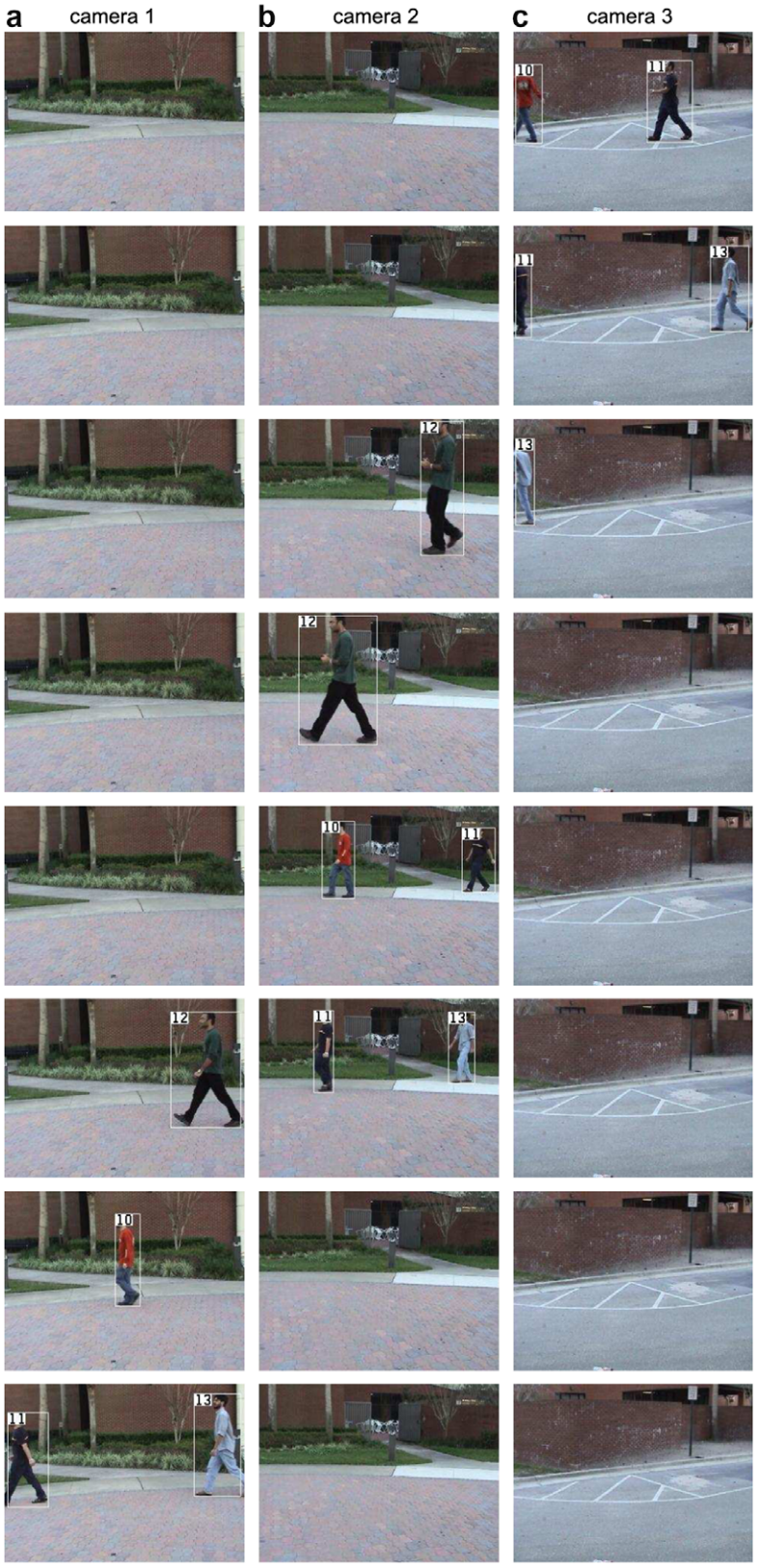

Fig. 15. Consistent labelling for camera setup 2. Rows 1 and 2: people enter Camera 3. Row 3: A new person enters Camera 2, note that he is given a previously unassigned label. Rows 4–8: People keep on moving across cameras. All persons retain unique labels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Table 3
The average normalized reconstruction errors for BTFs between observations of the same object, and between observations of different objects

Sequence No.	Average BTF reconstruction error (correct matches)	Average BTF reconstruction error (incorrect matches)
1	.0003	.0016
2	.0002	.0018
3	.0005	.0011

and three camera setups. A near real time implementation (5–10 frame/s on a 1.6 GHz machine) of the proposed multi-camera tracking approach was presented in a demo [16].

8. Conclusions

In this paper, we present space–time and appearance models for tracking objects across multiple non-overlapping cameras. These models are learned in a training phase. Using these models, we show that accurate tracking is possible even when observations of the objects are not available for relatively long periods of time due to non-overlapping camera views. The spatio-temporal cues used to constrain correspondences include inter-camera time intervals, location of exit/entrances, and velocities of objects. Moreover, for appearance matching, a novel method of modeling the change of appearance across cameras is presented. We show that given some assumptions, all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace. We also demonstrate empirically that even for real scenarios this subspace is low dimensional. The knowledge of camera parameters like focal length, aperture etc is not required for computation of the subspace of BTFs. The proposed system learns this subspace by using probabilistic principal component analysis on the BTFs obtained from the training data and uses it for the appearance matching. The space–time cues are combined with the appearance matching scheme in a ML framework for tracking. We have presented results on realistic scenarios to show the validity of the proposed approach.

Acknowledgment

This material is based upon work funded in part by the US Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Government.

Appendix A

Proof of Theorem 1. Let g_i and g_j be the radiometric response functions of cameras C_i and C_j, respectively. Also assume that for all $a, x \in \mathbb{R}$, $g_j(ax) = \sum_{u=1}^{m} r_u(a)s_u(x)$, where r_u and s_u are some arbitrary (but fixed) 1D functions, $1 \leqslant u \leqslant m$. Let f_{ij} be a brightness transfer function from camera C_i to camera C_j, then according to Eq. (11), f_{ij} is given as:

$$f_{ij} = g_j\left(wg_i^{-1}(\mathbf{B}_i)\right) = \left[g_j\left(wg_i^{-1}(B_i(1))\right), g_j\left(wg_i^{-1}(B_i(2))\right), \ldots, g_j\left(wg_i^{-1}(B_i(n))\right)\right]^{\mathrm{T}}$$

Since $g_j(ax) = \sum_{u=1}^{m} r_u(a)s_u(x)$, we may write f_{ij} as follows:

$$f_{ij} = \sum_{u=1}^{m} r_u(w)\left[s_u\left(g_i^{-1}(B_i(1))\right), s_u\left(g_i^{-1}(B_i(2))\right), \ldots, s_u\left(g_i^{-1}(B_i(n))\right)\right]^{\mathrm{T}} = \sum_{u=1}^{m} r_u(w)s_u\left(g_i^{-1}(\mathbf{B}_i)\right)$$

Thus, each brightness transfer function $f_{ij} \in \Gamma_{ij}$ can be represented as a linear combination of m vectors, $s_u(g_i^{-1}(\mathbf{B}_i))$, $1 \leqslant u \leqslant m$. Hence, the dimension of space Γ_{ij} is at most m.

References

[1] Special issue on video communications, processing, and understanding for third generation surveillance systems, in: Proceedings of the IEEE, vol. 89 (10), 2001.
[2] Q. Cai, J.K. Aggarwal, Tracking human motion in structured environments using a distributed camera system, IEEE Trans. Pattern Anal. Mach. Intell. 2 (11) (1999) 1241–1247.
[3] R. Collins, J. Lipton, T. Kanade, Introduction to the special section on video surveillance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000).
[4] R.T. Collins, A.J. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for cooperative multisensor surveillance, Proc. IEEE 89 (10) (2001) 1456–1477.
[5] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 564–575.
[6] S.L. Dockstader, A.M. Tekalp, Multiple camera fusion for multi-object tracking, in :IEEE Workshop on Multi-Object Tracking, 2001.
[7] R. Duda, P. Hart, D. Stork, Pattern Classification and Scene Analysis, Wiley, 2000.
[8] H. Farid, Blind inverse gamma correction, IEEE Trans. Image Process. 10 (10) (2001) 1428–1433.
[9] M.D. Grossberg, S.K. Nayar, Determining the camera response from images: what is knowable? IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1455–1467.
[10] M.D. Grossberg, S.K. Nayar, Modeling the space of camera response functions, IEEE Trans. Pattern Anal. Mach. Intell. 26 (10) (2004) 1272–1282.
[11] J. Hopcroft, R. Karp, An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. (1973).
[12] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.
[13] T. Huang, S. Russell, Object identification in a bayesian context, in: Proceedings of IJCAI, 1997.
[14] R. Jain, K. Wakimoto, Multiple perspective interactive video, in: IEEE International Conference on Multimedia Computing and Systems, 1995.
[15] O. Javed, M. Shah, Tracking and object classification for automated surveillance, in: Europen Conf. on Computer Vision, vol. 4, 2002, pp. 343–357.
[16] Omar Javed, Zeeshan Rasheed, Orkun Alatas, Asad Hakeem, Mubarak Shah, Real time surveillance in multiple non-overlapping cameras, in: Demonstration in CVPR, 2003.
[17] J. Kang, I. Cohen, G. Medioni, Continuous tracking within and across camera streams, in: IEEE Conf. Comput. Vision Pattern Recognition, 2003.
[18] V. Kettnaker, R. Zabih, Bayesian multi-camera surveillance, in: IEEE Conf. Comput. Vision Pattern Recognition, 1999, pp. 117–123.

[19] S. Khan, M. Shah, Consistent labeling of tracked objects in multiple cameras with overlapping fields of view, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003).
[20] C.G. Lambert, S.E. Harrington, C.R. Harvey, A. Glodjo, Efficient on-line nonparametric kernal density estimation, Algorithmica 25 (1999).
[21] L. Lee, R. Romano, G. Stein, Monitoring activities from multiple video streams: establishing a common coordinate frame, IEEE Trans. Pattern Recogn. Mach. Intell. 22 (8) (2000) 758–768.
[22] Q. Li, J. Racine, Nonparameteric estimation of distributions with categorical and continuous data, J. Multivariate Anal. 86 (2003) 266–292.
[23] D. Makris, T.J. Ellis, J.K. Black, Bridging the gaps between cameras, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
[24] S. Mann, R. Picard, Being undigital with digital cameras: extending dynamic range by combining differently exposed pictures, in: Proc. IS&T 46th Annual Conference, 1995.
[25] A. Mittal, L.S. Davis, M2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene, Int. J. Comput. Vis. 51 (3) (2003) 189–203.
[26] M. Oren, S.K. Nayar, Generalization of the Lambertian model and implications for machine vision, Int. J. Comput. Vis. 14 (3) (1995) 227–251.
[27] F. Porikli, Inter-camera color calibration using cross-correlation model function, in: IEEE Int. Conf. on Image Processing, 2003.
[28] A. Rahimi, T. Darrell, Simultaneous calibration and tracking with a network of non-overlapping sensors, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2004.
[29] K. Shafique, M. Shah, A non-iterative greedy algorithm for multi-frame point correspondence, in: IEEE Int. Conf. on Computer Vision, 2003.
[30] Y. Shan, H.S. Sahwney, R. Kumar, Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2005.
[31] C. Stauffer, Learning to track objects through un-observed regions, in: Proceedings of IEEE Workshop on Motion, 2005.
[32] M.J. Swain, D.H. Ballard, Indexing via color histograms, in: IEEE Int. Conf. on Computer Vision, 1990.
[33] Ting-Hsun, Chang, Shaogang Gong, Tracking multiple people with a multi-camera system, in: IEEE Workshop on Multi-Object Tracking, 2001.
[34] M.E. Tipping, C.M. Bishop, Probabilistic principal component analysis, J. R. Stat. Soc., B 61 (3) (1999) 611–622.
[35] M.P. Wand, M.C. Jones, Kernel Smoothing, Chapman & Hall, 1994.