

Pathnodes integration of standalone Particle Filters for people tracking on distributed surveillance systems

Roberto Vezzani, Davide Baltieri, and Rita Cucchiara

Dipartimento di Ingegneria dell'Informazione - University of Modena and Reggio Emilia, Via Vignolese, 905 - 41100 Modena - Italy
{roberto.vezzani,22218,rita.cucchiara}@unimore.it

Abstract. In this paper, we present a new approach to object tracking based on batteries of particle filter working in multicamera systems with non overlapped fields of view. In each view the moving objects are tracked with independent particle filters; each filter exploits a likelihood function based on both color and motion information. The consistent labeling of people exiting from a camera field of view and entering in a neighbor one is obtained sharing particles information for the initialization of new filtering trackers. The information exchange algorithm is based on pathnodes, which are a graph-based scene representation usually adopted in computer graphics. The approach has been tested even in case of simultaneous transitions, occlusions, and groups of people. Promising results have been obtained and here presented using a real setup of non overlapped cameras.

Key words: Pathnode, multicamera tracking

1 Introduction

Probabilistic methods are widely used for people tracking. In presence of occlusions and crowd, it is worth to keep more than one hypothesis of the current position of each tracked item: neighbor's body parts, noise, background objects, and other distracters often mislead the tracking system, resulting in a wrong state estimation and incorrect following. In a probabilistic framework the multiplicity of the state modes is faced with Mixtures of Gaussians or Particle Filtering. Particle Filtering has proven to be very successful for non linear and non Gaussian estimation problems such as people tracking in presence of occlusions and clutter. Thus, we propose a single camera tracking by particle filtering. This solution also allows a multicamera tracking framework as extension of a set of single camera systems.

2 Related work

Approaches to multicamera tracking can be generally classified into three main categories: *geometry-based*, *color-based*, and *hybrid* approaches [1]. The former

exploits geometrical relations and constraints between the different views to perform the consistent labeling process. They make use of known spatial information and relation among the set of available cameras. Two examples of this class are the work of Khan and Shah [2] and of Calderara et al [1]. Their approach is based on the computation of the so-called *Edges of Field of View*, i.e. the lines delimiting the field of view of each camera and, thus, defining the overlapped regions. Geometry-based approaches have been proved to be very reliable and effective, but they require at least a partial overlap between camera fields of view. In non overlapped camera frameworks, color-based approaches can be successfully applied, since the matching is essentially based on the color of the tracks. For example, in [3] a color space invariant to illumination changes is proposed and histogram-based information at low (texture) and mid (regions and blobs) level are adopted. Conversely, the work in [4] uses stereoscopic vision to match tracks, but when this matching is not sufficiently reliable, color histograms are used to solve ambiguities. Finally, Hybrid approaches, belonging to the third class, mix information about the geometry and the calibration with those provided by the visual appearance. These last methods are based on probabilistic information fusion [5] or on Bayesian Belief Networks (BBN) [6]. Our approach belongs to the hybrid class, since it makes use of a weak calibration in order to locate each camera in a global map of the environment. Geometrical information are used to make predictions as in [7], but they are not enough since the cameras are not overlapped. Then, color information are used to perform accurate track matches in different camera field of view.

3 Particle filter for Single Camera Tracking

Goal of a single camera tracking system is to estimate the state \mathbf{x}_t at frame t given the set of observations $\{\mathbf{z}_{1..t}\}$. At this stage we assume for simplicity that each tracked object is moving regardless of other people in the scene. Thus, an independent tracker is applied for each object in order to estimate its state \mathbf{x} , which we assume to be composed by the coordinates of the gravity center (x_c, y_c) . In a probabilistic framework, \mathbf{x} is a random variable, with the associated probability density function $p(\mathbf{x}|\mathbf{z})$ which we are interested on. The most likely position $\hat{\mathbf{x}}$ of each tracked person can be estimated at time t from the relative pdf as $\hat{\mathbf{x}}_t = E(\mathbf{x}_t)$. To this aim we adopted a generic particle filtering technique; for the sake of completeness we report here the base equations using the notation proposed by Arulampalam et al in their famous tutorial [8].

Let $\{\mathbf{x}^i, w^i\}_{i=1..N}$ be a characterization of the posterior probability $p(\mathbf{x}|\mathbf{z})$, where $\{\mathbf{x}^i, i = 1..N\}$ is the set of support points (particles) with their weights $\{w^i, i = 1..N\}$. The weights are normalized to add up to one. By means of this set of weighted particles the posterior probability can be approximated as:

$$p(\mathbf{x}|\mathbf{z}) \approx \sum_{i=1}^N w^i \delta(\mathbf{x} - \mathbf{x}^i), \quad w_t^i \propto w_{t-1}^i \cdot p(\mathbf{z}_t|\mathbf{x}^i) \quad (1)$$

Table 1. Pseudo-code of resampling and particle filter algorithms

RESAMPLING ALGORITHM <ul style="list-style-type: none"> – Initialize the CDF: $c_1 = 0$ – FOR $i = 2 : N_s$ <ul style="list-style-type: none"> • Construct CDF: $c_i = c_{i-1} + w_k^i$ – Start at the bottom of the CDF: $i = 1$ – Draw a starting point: $u_1 \sim \mathcal{U}[0, N_s^{-1}]$ – FOR $j = 1 : N_s$ <ul style="list-style-type: none"> • Move along the CDF: $u_j = u_i + N_s^{-1}(j - 1)$ • WHILE $u_j > c_i$ * $i = i + 1$ • Assign sample: $\mathbf{x}_k^{j*} = \mathbf{x}_k^i$ • Assign weight: $w_k^j = N_s^{-1}$ 	GENERIC PARTICLE FILTER <ul style="list-style-type: none"> – FOR $i = 1 : N_s$ <ul style="list-style-type: none"> • Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ • Assign the particle a weight, w_k^i according to Eq. (1) – Normalize w_k^i such that $\sum_i w_k^i = 1$ – Calculate \widehat{N}_{eff} – IF $\widehat{N}_{eff} < N_T$ <ul style="list-style-type: none"> • RESAMPLING ALGORITHM
---	---

where we adopt as importance density the prior $p(\mathbf{x}_t | \mathbf{x}_{t-1}^i)$, and the resampling step is executed only if the measure of degeneracy $\widehat{N}_{eff} = \left[\sum_{i=1}^N (w^i)^2 \right]^{-1}$ is lower than a threshold N_T . The particle filter tracking can be described by the two pseudo-code algorithms reported in Tab. 1 (from Algorithms 2 and 3 in [9]). The adopted likelihood function and the new occlusion-based process model, instead, are fully described in the following.

3.1 The likelihood function

Similar to work of Li et al [10], for each object we store and keep updated a set of three appearance models $AM_t = \{am_t^1 \doteq F_t, am_t^2 \doteq A_t, am_t^3 \doteq D_t\}$. We assume that each pixel is independent of each other. The fixed model F_t contains the appearance of the tracked object stored at the initialization phase. The Adaptive model A_t stores the mean of N object appearances sampled at regular time steps. Finally, the Dynamic model D_t is estimated averaging D_{t-1} with the current frame. All these models can take advantage of a foreground segmentation of the current frame. Since in the regarded application we make use of fixed cameras, we process every frame with a background subtraction system, SAKBOT [11]. Only the image points classified as foreground pixels concur to the model estimation. Calling I_t the current frame, the model update equations are:

$$AM_t = \{am^1, am^2, am^3\} = \begin{cases} am^1 = F_t(\mathbf{x}) = F(\mathbf{x}) = I_{t_1}(\mathbf{x} - \hat{\mathbf{x}}_t) \\ am^2 = A_t(\mathbf{x}) = \frac{1}{N} \cdot \sum_{j=1}^N I_{t-j\Delta t}(\mathbf{x} - \hat{\mathbf{x}}_{t-j\Delta t}) \\ am^3 = D_t(\mathbf{x}) = \frac{1}{2} \cdot D_{t-1}(\mathbf{x}) + I_t(\mathbf{x} - \hat{\mathbf{x}}_t) \end{cases} \quad (2)$$

where I_{t_i} is the first frame of the i -th object, i.e. when it is entered the scene; the notation $\mathbf{x} - \hat{\mathbf{x}}_t$ is used to indicate the frame by frame alignment of the model using the estimated position of the target $\hat{\mathbf{x}}_t$.

On the whole, the state $\{\hat{\mathbf{x}}, AM\}$ of a tracked object is composed by $\hat{\mathbf{x}}$, i.e., the position of the center of mass in the image plane (estimated by particle filtering), and its appearance AM (updated using Eq. 2).

With the likelihood function we can estimate $p(\mathbf{z}_t|\mathbf{x}_t)$, i.e., how likely a particular object position is to produce the current frame. Practically, we have to compare the estimated models for each object at the previous frame AM_{t-1} with the current image I_t . A pixel-wise comparison usually leads to errors if the model is not exactly aligned with the current frame, since a pixel-by-pixel distance is not a monotonic function. Therefore the distance measure between the model and the current observation is based on aggregate functions like color histograms.

Let be $RR_d(I, \mathbf{x})$ a rectangular region of the image I , centered around the point \mathbf{x} , fixed shape ratio and scale factor d derived from the previous background suppression phase. Let $H(\cdot)$ be the histogram of the image argument. Then, at sampling time one of the three appearance models $am^j \in AM_t$ is randomly selected for each particle. The likelihood value $p(\mathbf{z}_t|\mathbf{x}_t)$ is extracted from a zero-mean normal distribution using the Bhattacharyya distance $\Phi(\cdot)$ between the color histogram from the current image $H(RR_d(I_t, \hat{\mathbf{x}}_t))$ and the selected model histogram $H(am^j)$ as in Eq. 3

$$p(\mathbf{z}_t|\mathbf{x}_t^i) = p(\mathbf{z}_t|\mathbf{x}_t^i, am^j) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{\Phi(H(RR_d(I_t, \hat{\mathbf{x}}_t)), H(am^j))}{2\sigma^2}} \quad (3)$$

3.2 The Process Model

The motion of a person in a scene is difficult to predict and is seldom linear, in particular if the object position is measured in image coordinates. Therefore, usually the random walk equations are adopted for the state prediction step [12]. Our approach is slightly different and takes into account the previous movements of the person in addition to the Gaussian noise. We estimate the current person's speed $\hat{\mathbf{v}}_t$ by means of an exponentially weighted average schema:

$$\hat{\mathbf{v}}_t = \beta\hat{\mathbf{v}}_{t-1} + (1 - \beta)(\hat{\mathbf{x}}_t - \hat{\mathbf{x}}_{t-1}) \quad (4)$$

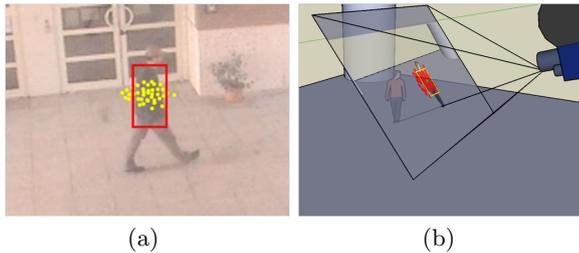


Fig. 1. Graphical representation of the tracked state through particle filter. Dots are the particles positions and the rectangle is the ROI for the likelihood computation and model update

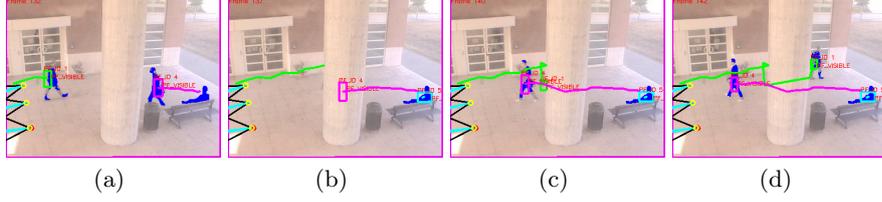


Fig. 2. Example of person tracking during a strong occlusion

where β has been empirically set to $\frac{2}{3}$.

$$\tilde{\mathbf{x}}_t^i = \mathbf{x}_{t-1}^i + \hat{\mathbf{v}}_{t-1} + \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}_2). \quad (5)$$

The Gaussian noise of Eq. 5 should take into account the non linear nature of the human motion. Furthermore the tracker can be misled by occlusions and shape changes. Thus, the Gaussian noise should be large enough to manage the unpredictable changes in speed and direction, but it does not set to naught the linear prediction. To this aim we propose a spherical covariance with a dynamic parameter σ_t which depends on the likelihood score computed at the previously estimated position $\hat{\mathbf{x}}_{t-1}$:

$$\sigma_t = \frac{\sigma_0}{(1 + \alpha p(\mathbf{z}_{t-1} | \hat{\mathbf{x}}_{t-1}))}. \quad (6)$$

where σ_0 is mandatory since the variance should be greater than zero and α is a predefined constant. During occlusions or quick motion changes the distance function computed in the estimated position grows; increasing the noise term. This will assure that particles will be much spread in the next step. Figure 2 reports an example of long-lasting occlusions. Even that, the tracker system can manage the label assignments after the occlusions thanks to the dynamic nature of the noise term.

3.3 Initialization

As above mentioned, each object is independently considered and tracked; every time a person enters the scene a new particle filter should be created and initialized. To this aim the foreground mask extracted from the background subtraction algorithm is labeled and unassigned blobs are classified as new objects. A blob is unassigned if no particle has required that blob for the likelihood function estimation. To avoid wrong assignments between new blobs and particles associated with other objects, a blob is associated to a particle (and then eliminated from the list of new objects to be tracked) if the correspondent Bhattacharyya distance is under a threshold. The initial position $\hat{\mathbf{x}}_0$ of the object is set equal to the blob gravity center and the three appearance models are initialized with the blob appearance.

4 Multi Camera Tracking

Let us suppose to have a wide space monitored by a set of non overlapped fixed cameras. If the fields of view of the cameras are close enough, it is plausible that a person exiting a camera field of view will soon appear on the field of view of a neighbor camera. Moreover, if the illumination conditions and the camera type are the same, a person will generate similar color histograms on different views. Information stored in the particle filter used to track a person in the source camera can be effectively used to initialize the tracking system of the destination camera.

We propose to exploit computer graphic techniques to manage the multicamera tracking problem. Interactions between different scenes and the movements of avatars are usually managed with graph models by the computer graphic and virtual reality communities. All the possible avatar positions (or rooms) are represented by nodes and the connecting arcs refers to allowed paths. The sequence of visited nodes is called *pathnodes*. A weight can be associated to each arc in order to give some measures on it, such as the duration, the likelihood to be chosen with respect to other paths, and so on. We empirically set the arc weights, but a learning phase can be established to automatically compute them.

In Fig. 3(a) a sample test bed is depicted; three cameras are installed and the correspondent fields of view are highlighted. A person walking in the represented scene can stay inside the field of view of the same camera, can move toward another one trough a transition zone or can definitely exits the area. According with the allowed paths, the overall area can be divided into three region types: A. Visible regions, corresponding to camera's fields of view; B. Non visible regions, i.e., transition areas between two or more visible regions; C. Forbidden or exit regions.

In particular we are interested to identify the boundaries of these regions. People inside visible regions can be detected and tracked by the corresponding single camera system; once they exit that region to enter in a non visible region, they will likely appear again in another visible region. The exit and the entry points are located on the boundaries between visible and non visible regions. If a person exits the entire scene walking through a forbidden region boundaries he will not be seen again. In other words, the state of a person can be estimated by the single camera tracker without exchanging information with the other systems during all his permanence on the field of view of a camera. If the person reach a boundary with a non visible region, his state should be propagated to the neighbor cameras in order to consistently label the person once he will appear again in another visible area. Finally, if the person crosses a forbidden or exit region boundary his state can be discarded since it is improbable that he will reappear in the scene (Fig 3(b)). Let us define the following symbols:

- $\{C^1 \dots C^K\}$: set of K installed cameras;
- $FoV(C^k)$: visible regions, i.e, the field of view of a camera C^k ;
- $\{NVR^1 \dots NVR^L\}$: non visible regions;
- $\{FR^1 \dots FR^H\}$: forbidden or exit regions;

- $R^l \cap R^k$: boundary between two regions;

Through the Process Model described in Section 3.2 we can estimate $p(\mathbf{x}_{t+\Delta t}^i | \mathbf{x}_t^i)$ only for object positions inside the same camera field of view. Our goal is to extend that model for neighbor cameras, evaluating the probability of moving toward another camera FoV:

$$p(\mathbf{x}'_{t+\Delta t} | \mathbf{x}_t^i) \quad (7)$$

where \mathbf{x}' and \mathbf{x} are referred to two different FoV regions.

4.1 Pathnode based Scene Representation

Due to preferential paths, perspective transformations, and scene nature, the probability of Eq. 7 cannot be expressed in analytical form. Following computer graphic techniques, we then generate a graph-based representation which can be considered an approximation of it. Actually, the domain of Eq. 7 can be restricted to the boundaries between visible and non visible regions.

We introduce the Pathnode based Scene Representation, composed by:

- $\Psi = \{\psi_i\}$: set of nodes $\psi_i \in \{FoV(C^k) \cap NVR^l, k = 1 \dots K, l = 1 \dots L\}$. Each node ψ_i is associated to an object position \mathbf{x}_{ψ_i} on the boundary between $FoV(C^k)$ and NVR^l ;
- $\Gamma = \{\gamma_{i,j}\}$: set of weighted arcs $\gamma_{i,j} \subset \{\Psi \times \Psi\}$, corresponding to a possible path from two nodes ψ_i and ψ_j that people can follow. The weight $c(\gamma_{i,j}) \in \mathbb{R}$ of each arc is proportional to the use frequency of the set of paths it represents:

$$c(\gamma_{i,j}) \propto p(\mathbf{x}_{t+\Delta t} = \mathbf{x}_{\psi_j} | \mathbf{x}_t = \mathbf{x}_{\psi_i}), \quad \sum_i c(\gamma_{i,j}) = 1. \quad (8)$$

4.2 Hand-off management

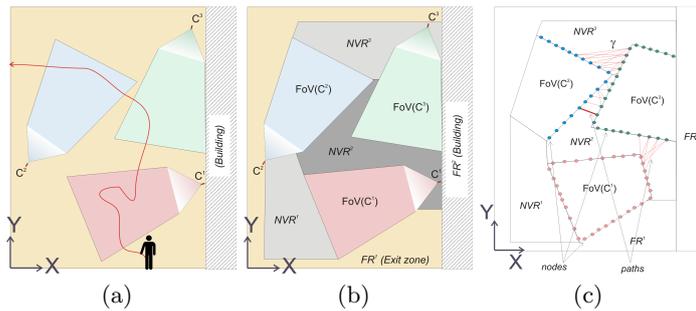


Fig. 3. A sample surveillance system. (a) FoV of three cameras, (b) region classification and graph-based representation with pathnodes

Each time a tracked person approaches a boundary of the visible region, a new particle filter tracker is created on the cameras connected by at least one arc. Let us suppose that a person is exiting the field of view of the camera C^s . If a camera C^d is adjacent to C^s (i.e., at least one arc $\gamma_{i,j}$ connects one node of C^s to one node of C^d) a new particle filter is created in C^d , sharing the same appearance models and the same identifier of the source one, up to color transformations as afterwards described in Section 5.

The initial set of particles $\{\mathbf{x}'^i, w'^i\}_{i=1..N}$ is sampled from the proposal distribution $q(\mathbf{x}'_t | \mathbf{x}_{t-\Delta t})$, where the position \mathbf{x}' in the destination camera C^d is a function of the particle positions in the source one C^s .

$$q(\mathbf{x}'_t | \mathbf{x}_{t-\Delta t}^i) = \sum_{\psi_j \in C^d} \alpha_j \cdot \mathcal{N}(\mathbf{x}_{\psi_j}, \sigma) \quad (9)$$

$$\alpha_j = w^i \cdot \sum_{\psi_k \in C^s} c(\gamma_{k,j}) \cdot \delta_{\psi_k}(\mathbf{x}_{t-\Delta t}^i) \quad (10)$$

$$\delta_{\psi_k}(\mathbf{x}^i) = \begin{cases} 1 & \text{if } \|\mathbf{x}^i, \mathbf{x}_{\psi_k}\| \leq \beta \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

5 Inter-camera Color Transformation

If the cameras are of different types or the illumination conditions are not uniform the main hypothesis of color portability is no more valid. Then, an inter-camera color calibration is required. To this aim different algorithms have been proposed, ranging from simple linear transformations to more generic solutions such as lookup tables. The two sample frames of Fig.4(a) show a situation in which the color calibration problem is particularly marked.



Fig. 4. a. A common problem: different views have different colors; b. Patterns used for the color calibration.

For computational reasons we adopt a fast linear transformation, which consist in finding a matrix \mathbf{M} able to model the space changes. In [13] Roullout shows that the linear algebraic model is the most natural model under the usual assumptions, providing the detailed mathematical justification. A set of couple of correspondent colors have been manually extracted from captured images of a

	#frames	FPS	#people	Correct	FN	FP	Double Assignment	Id exchange	Handoff errors	Precision	Recall
Video1	301	13,3	9	536	34	171	135	1	0,25	76%	94%
Video2	837	11,2	14	1119	75	134	90	4	0,44	89%	94%
Video3	500	11,5	12	547	35	48	48	1	0,00	92%	94%
Video4	1020	13,4	16	707	134	68	34	0	0,50	91%	84%
Video5	769	10,6	28	1547	161	196	30	1	0,00	89%	91%
Video6	633	12,1	10	435	113	3	2	0	0,30	99%	79%
	4060	12	89	4891	552	620	339	7	0,25	89%	90%

Fig. 5. Quantitative performance evaluation of the tracking system

color pattern as in Fig. 4(b). The matrix \mathbf{M} is thus obtained with a least squares optimization.

6 Experimental results

The proposed method has been implemented in C++, partially using the OpenCV libraries [14] and the Imagelab processing libraries. From a computational point of view, the system with two running cameras is able to process 10 frames per second if the number of people in the scene is about ten, i.e., two background segmentation processing and ten particle filters are simultaneously working.

To test the application, we used some videos from the surveillance system installed in our department. The system is composed by four cameras: three of them are fixed and the fourth is a PTZ dome camera. Three different scenarios obtained selecting couples of non overlapped views have been defined and used to generate the data set. A graphical interface allows to define the global map, the fields of view and the nodepaths.

We captured and tested the application on 6 videos of different complexity and we collected results in the table of Fig. 5. For each sequence we counted the number of appeared people. For each frame we identified the number of correct track assignments and the number of False Positives (FP) and False Negatives (FN) and their sum over the whole video are reported in the table. The mean precision and recall are 89% and 90% respectively for the in-camera tracking, while the consistent labeling task reaches a 80% precision rate. The different types of error are also reported in the table. As it is possible to see, most of the FP errors are due to double assignments, which are due to more than one particle filter being created for the same object, as a consequence of segmentation errors arising from the background suppression phase. Finally, there is an ID exchange each time the system fails to disambiguate after an interaction or a mutual occlusion among two or more people.

7 Conclusions and future work

In this paper we propose an integrated framework for in-camera and multi camera people tracking. First of all, a single camera particle filtering approach is proposed, in order to keep more than one hypothesis on the position of each tracked

person. Such a condition is required in order to manage occlusions, group of people and color ambiguity. Experimental results on our real in-campus surveillance system show the validity and the robustness of the approach. Furthermore, the required computational load allows a real time implementation. Moreover, a new approach to multi camera tracking is the main contribution of this paper. Information embedded on each single camera tracker are diffused to other cameras, by means of a graph representation of possible connecting paths. The approach is very promising and the results on real datasets are good. Even so, color features are not enough distinctive and view invariant to be reliably transferred among different cameras. We proposed an inter-camera color transformation to reduce this problem, but it is still inadequate. We are currently working on finding more appropriate features and invariants.

References

1. Calderara, S., Cucchiara, R., Prati, A.: Bayesian-competitive consistent labeling for people surveillance. *IEEE Trans. on PAMI*, 30, 354–360 (2008)
2. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. on PAMI*, 25, 1355–1360 (2003)
3. Li, J., Chua, C., Ho, Y.: Color based multiple people tracking. In: *Proc. of IEEE Intl Conf. on Control, Automation, Robotics and Vision*. pp. 309–314 (2002)
4. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for easyliving. In: *Proc. of IEEE Intl Workshop on Visual Surveillance*. pp. 3–10 (2000)
5. Kang, J., Cohen, I., Medioni, G.: Continuous tracking within and across camera streams. In: *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*. Volume 1. pp. 267–272 (2003)
6. Chang, S., Gong, T.H.: Tracking multiple people with a multi-camera system. In: *Proc. of IEEE Workshop on Multi-Object Tracking*. pp. 19–26 (2001)
7. Yue, Z., Zhou, S., Chellappa, R.: Robust two-camera tracking using homography. In: *Proc. of IEEE Intl Conf. on Acoustics, Speech, and Signal Processing*. pp. 1–4 (2004)
8. Arulampalam, S., Maskell, S., Gordon, N.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50, 174–188 (2002)
9. Doucet, A.: On sequential simulation-based methods for Bayesian filtering. Technical report (1998)
10. Li, A., Jing, Z., Hu, S.: Robust observation model for visual tracking in particle filter. *AEU - International Journal of Electronics and Communications*, 61, 186–194 (2007)
11. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. on PAMI*, 25, 1337–1342 (2003)
12. Feller, W.: *An Introduction to Probability Theory and Its Applications*, Volume 1. Wiley (1968)
13. Roullot, E.: A unifying framework for color image calibration. In: *Proc. of IWSSIP 2008*. 97–100 (2008)
14. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA (2008)