

RESEARCH

Open Access

# Multi-camera multi-object voxel-based Monte Carlo 3D tracking strategies

Cristian Canton-Ferrer\*, Josep R Casas, Montse Pardàs and Enric Monte

## Abstract

This article presents a new approach to the problem of simultaneous tracking of several people in low-resolution sequences from multiple calibrated cameras. Redundancy among cameras is exploited to generate a discrete 3D colored representation of the scene, being the starting point of the processing chain. We review how the initiation and termination of tracks influences the overall tracker performance, and present a Bayesian approach to efficiently create and destroy tracks. Two Monte Carlo-based schemes adapted to the incoming 3D discrete data are introduced. First, a particle filtering technique is proposed relying on a volume likelihood function taking into account both occupancy and color information. Sparse sampling is presented as an alternative based on a sampling of the surface voxels in order to estimate the centroid of the tracked people. In this case, the likelihood function is based on local neighborhoods computations thus dramatically decreasing the computational load of the algorithm. A discrete 3D re-sampling procedure is introduced to drive these samples along time. Multiple targets are tracked by means of multiple filters, and interaction among them is modeled through a 3D blocking scheme. Tests over CLEAR-annotated database yield quantitative results showing the effectiveness of the proposed algorithms in indoor scenarios, and a fair comparison with other state-of-the-art algorithms is presented. We also consider the real-time performance of the proposed algorithm.

## 1 Introduction

Tracking multiple objects and keeping record of their identities along time in a cluttered dynamic scene is a major research topic in computer vision, basically fostered by the number of applications that benefit from the retrieved information. For instance, multi-person tracking has been found useful for automatic scene analysis [1], human-computer interfaces [2], and detection of unusual behaviors in security applications [3].

A number of methods for camera-based multi-person 3D tracking have been proposed in the literature [4-7]. A common goal in these systems is robustness under occlusions created by the multiple objects cluttering the scene when estimating the position of a target. Single-camera approaches [8] have been widely employed, but they are vulnerable to occlusions, rotation, and scale changes of the target. In order to avoid these drawbacks, multi-camera tracking techniques exploit spatial redundancy among different views and provide 3D information at the actual scale of the objects in the real world. Integration

of data extracted from multiple cameras has been proposed in terms of a fusion at feature level as image correspondences [9] or multi-view histograms [10] among others. Information fusion at data or raw level has been achieved by means of voxel reconstructions [11], polygon meshes [12], etc.

Most multi-camera approaches rely on a separate analysis of each camera view, followed by a feature fusion process to finally generate an output. Exploiting the underlying epipolar geometry of a multi-camera setup toward finding the most coherent feature correspondence among views was first tackled by Mikič et al. [13] using algebraic methods together with a Kalman filter, and further developed by Focken et al. [14]. Exploiting epipolar consistency within a robust Bayesian framework was also presented by Canton-Ferrer et al. [9]. Other systems rely on detecting semantically relevant patterns among multiple cameras to feed the tracking algorithm as done in [15] by detecting faces. Particle filtering (PF) [16] has been a commonly employed algorithm because of its ability to deal with problems involving multi-modal distributions and non-linearities. Lanz et al. [10] proposed a multi-camera PF tracker exploiting foreground and color

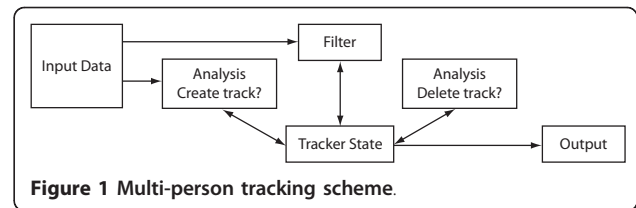
\* Correspondence: [cristian.canton@gmail.com](mailto:cristian.canton@gmail.com)  
Image and Video Processing Group, Technical University of Catalonia,  
Barcelona, Spain

information, and several contributions have also followed this path: [4,7]. Occlusions, being a common problem in feature fusion methods, have been addressed in [17] using HMM to model the temporal evolution of occlusions within a PF algorithm. Information about the tracking scenario can also be exploited toward detecting and managing occlusions as done in [18] by modeling the occluding elements, such as furniture, in a training phase before tracking. It must be noted that, in this article, we assume that all cameras will be covering the area under study. Other approaches to multi-camera/multi-person tracking do not require maximizing the overlap of the field of view of multiple cameras, leading to the non-overlapped multi-camera tracking algorithms [19].

Multi-camera/multi-person tracking algorithms based on a data fusion before doing any analysis was pioneered by Lopez et al. [20] by using a voxel<sup>a</sup> reconstruction of the scene. This idea was further developed by the authors in [5,21] finally leading to the present article. Up to our knowledge, this is the first approach to multi-person tracking exploiting data fusion from multiple cameras as the input of the algorithms. In this article, we first introduce a methodology to multi-person tracking based on a colored voxel representation of the scene as the start of the processing chain. The contribution of this article is twofold. First, we emphasize the importance of the initiation and termination of tracks, usually neglected in most tracking algorithms, that has indeed an impact on the performance of the overall system. A general technique for the initiation/termination of tracks is presented. The second contribution is the filtering step where two techniques are introduced. The first technique applies PF to input voxels to estimate the centroid of the tracked targets. However, this process is far from real-time performance and an alternative, that we call Sparse Sampling (SS). SS aims at decreasing computation time by means of a novel tracking technique based on the seminal PF principle. Particles no longer sample the state space but instead a magnitude whose expectancy produces the centroid of the tracked person: the surface voxels. The likelihood evaluation relying on occupancy and color information is computed on local neighborhoods, thus dramatically decreasing the computation load of the overall algorithm. Finally, effectiveness of the proposed techniques is assessed by means of objective metrics defined in the framework of the CLEAR [22] multi-target tracking database. Computational performance is reviewed toward proving the real-time operation of the SS algorithms. Fair comparisons with state-of-the-art methods evaluated using the same database are also presented and discussed.

## 2 Tracker design methodology

Typically, a multi-target tracking system can be depicted as in Figure 1 and comprises a number of elementary



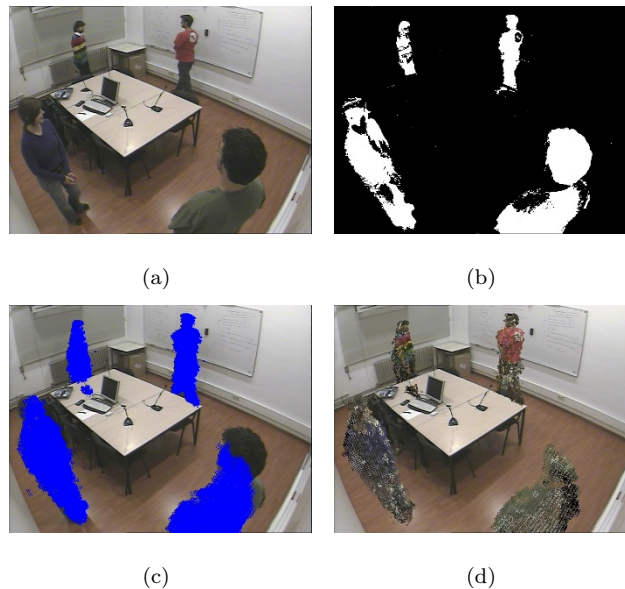
modules. Although most articles present techniques that contribute to filtering module, the overall architecture is rarely addressed assuming that some blocks are already available. In this section, this scheme will be analyzed and some proposals for each module will be presented. The filtering step, being our major contribution, will be addressed in a separate section.

### 2.1 Input and output data

When addressing the problem of multi-person tracking within a multi-camera environment, a strategy about how to process this information is needed. Many approaches perform an analysis of the images separately, and then combine the results using some geometric constraints [10]. This approach is denoted as an information combination by fusion of decisions. However, a major issue in this procedure is dealing with occlusion and perspective effects. A more efficient way to combine information is data fusion [23]. In our case, data fusion leads to a combination of information from all images to build up a new data representation, and to apply the algorithms directly on these data. Several data representations aggregating the information of multiple views have been proposed in the literature such as voxel reconstructions [11,24], level sets [25], polygon meshes [12], conexels [26], depth maps [27], etc. In our research, we opted for a colored voxel representation due to both its fast computation and accuracy.

For a given frame in the video sequence, a set of  $N_C$  images are obtained from the  $N_C$  cameras (see a sample in Figure 2(a)). Each camera is modeled using a pinhole camera model based on perspective projection with camera calibration information available. Foreground regions from input images are obtained using a segmentation algorithm based on Stauffer-Grimson's background learning and subtraction technique [28] as shown in Figure 2(b).

Redundancy among cameras is exploited by means of a Shape-from-Silhouette (SfS) technique [11]. This process generates a discrete occupancy representation of the 3D space (voxels). A voxel is labeled as foreground or background by checking the spatial consistency of its projection on the  $N_C$  segmented silhouettes, and finally obtaining the 3D binary reconstruction shown in Figure 2(c). We will denote this raw voxel reconstruction as  $\mathcal{V}$ . The visibility of a surface voxel onto a given camera is



**Figure 2 Input data generation example.** (a) A sample of the original images. (b) Foreground segmentation of the input images employed by the SfS algorithm. (c) Example of the binary 3D voxel reconstruction. (d) The final colored version shown over a background image.

assessed by computing the discrete ray originating from its optical center to the center of this voxel using Bresenham's algorithm and testing whether this ray intersects with any other foreground voxel. The most saturated color among pixels of the set of cameras that see a surface voxel is assigned to it. A colored representation of surface voxels of the scene is obtained, denoted as  $\mathcal{V}^C$ . An example of this process is depicted in Figure 2(d). It should be taken into account that, without loss of generality, other background/foreground and 3D reconstruction algorithms may be used to generate the input data to the tracking algorithm presented in this article.

The resulting colored 3D scene reconstruction is fed to the proposed system that assigns a tracker to each target and the obtained tracks are processed by a higher semantic analysis module. Information about the environment (dimensions of the room, furniture, etc.) allows assessing the validity of tracked volumes and discarding false volume detections.

Finally, the output of the overall tracking algorithm will be a number of hypotheses for the centroid position of each of the targets present in the scene.

## 2.2 Tracker state and filtering

One of the major challenges in multi-target tracking is the estimation of the number of targets and their positions in the scene, based on a set of uncertain observations. This issue can be addressed from two perspectives. First, extending the theory of single-target algorithms to multiple targets. This approach defines the working state space  $\mathcal{X}$  as the concatenation of the positions of all  $N_T$

targets as  $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_T}]$ . The difficulty here is the time variant dimensionality of this space. Monte Carlo approaches, and specifically PF approaches, to this problem have to face the exponential dependency between the number of particles required by the filter and the dimension of  $\mathcal{X}$ , turning out to be computationally infeasible. Recently, a solution based on random finite sets achieving linear complexity has been presented [29].

Multi-target tracking can also be tackled by tracking each target independently, that is to maintain  $N_T$  trackers with a state space  $\mathcal{X}_i = \mathbf{x}_i$ . In this case, the system attains a linear complexity with the number of targets, thus allowing feasible implementations. However, interactions among targets must be modeled in order to ensure the most independent set of tracks. This approach to multi-person tracking will be adopted in our research.

## 2.3 Track initiation and termination

A crucial factor in the performance of a tracking system is the module that addresses the initiation and termination of tracks. The initiation of a new tracker is independent of the employed filtering technique and only relies on the input data and the current state (position) of the tracks in the scene. On the other hand, the termination of a new tracking filter is driven by the performance of the tracker.

The initialization of a new filter is determined by the correct detection of a person in the analyzed scene. This process is crucial when tracking, and its correct operation will drive the overall system's accuracy. However,

despite the importance of this step, little attention is paid to it in the design of multi-object trackers in the literature. Only few articles explicitly mention this process such as [30] that employs a face detector to detect a person or [31] that uses scout particle filters to explore the 3D space for new targets. Moreover, it is assumed that all targets in the scene are of interest, i.e., people, not accounting for spurious objects, i.e., furniture, shadows, etc. In this section, we introduce a method to properly handle the initiation and termination of filters from a Bayesian perspective.

### 2.3.1 Track initiation criteria

The 3D input data  $\mathcal{V}$  fed to the tracking system is usually corrupted and presents a number of inaccuracies such as objects not reconstructed, mergings among adjacent blobs, spurious blobs, etc. Hence, defining a track initialization criterium based solely on the presence of a blob might lead to poor performance of the system. For instance, objects such as furniture might be wrongly detected as foreground, reconstructed and tracked. Instead, a classification of the blobs based on a probabilistic criteria can be applied during this initialization process aiming at a more robust operation. Training of this classifier is based on the development set of the used database, together with the available ground truth describing the position of the tracked objects.

Let  $\mathbf{X}^{\text{GT}} = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_{\text{GT}}}\}$  be the ground truth positions of the  $N_{\text{GT}}$  targets present in the scene of the development set at a given instant. Once the reconstruction  $\mathcal{V}$  is available, a connected component analysis is performed over these data thus obtaining a set of  $K$  disjoint components,  $\mathcal{C}_i$ , fulfilling:

$$\mathcal{V} = \bigcup_{i=1}^K \mathcal{C}_i. \quad (1)$$

We will consider the region of influence of a target with centroid  $\mathbf{x}$  as the ellipsoid  $\mathcal{E}(\mathbf{x}, \gamma)$  with axis size  $\mathbf{s} = (s_x, s_y, s_z)$  centered at  $\mathbf{c}$ .

A mapping is defined such that for every  $\mathbf{x}_j \in \mathbf{X}^{\text{GT}}$  a component  $\mathcal{C}_i$  is assigned. Let us denote  $[\mathbf{x}]_{\{x,y,z\}}$  as the  $x$ ,  $y$  or  $z$  coordinate of vector  $\mathbf{x}$ . The assignation process is defined as follows: first, a region of influence  $\mathcal{E}(\mathbf{x}_j, \mathbf{s})$  with size  $\mathbf{s} = (s_x, s_y, [\mathbf{x}_j]_z)$  centered at  $\mathbf{c} = \mathbf{x}_j$  is placed in the 3D space. The radii  $s_x$  and  $s_y$  are chosen to contain an average person,  $s_x = s_y = 30$  cm. Let us define the operator  $|\cdot|$  applied to a volume as the number of non-zero voxels contained in it. Then, the assignation is defined as

$$\mathbf{x}_j \rightarrow \arg \max_i |\mathcal{E}(\mathbf{x}_j, \mathbf{s}) \cap \mathcal{C}_i|, \quad (2)$$

that is to assign  $\mathbf{x}_j$  to the component with the largest volume enclosed in the region of influence. It must be noted that some  $\mathbf{x}_j$  might not have any  $\mathcal{C}_i$  associated due to a wrong segmentation or faulty reconstruction of the target. Moreover, the set of components not associated to any ground truth position can be identified as spurious objects, reconstructed shadows, etc.

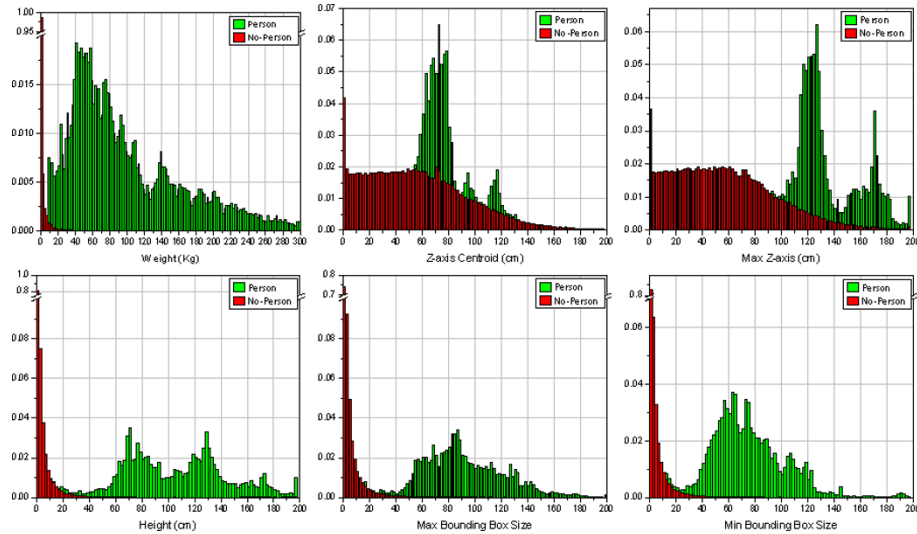
Finally, we have grouped the set of connected components  $\mathcal{C}_i$  in two categories: person and non-person. A set of features are extracted from each of these components, thus conforming the characteristics that will be used to train a person/no-person binary classifier. This set of extracted features is described in Table 1.

In order to characterize the objects to be tracked and to decide the best classifier system, we have performed an exploratory data analysis [32], which will allow us to contrast the underlying hypotheses of the classifiers with the actual data. Histograms of these features are computed as shown in Figure 3 and scatter plots depicting the cross dependencies among all features are

**Table 1 Features employed by the person/no-person classifier where magnitude  $|\mathcal{V}|_{\{x,y,z\}}$  denotes the  $x$ ,  $y$ , or  $z$  coordinates of voxel  $\mathcal{V}$**

Feature	Expression
Weight	$ \mathcal{C}_i  s_v^3 \rho \quad \rho = 1.1 \text{ [gr/cm}^3\text{]}$
Centroid (z-axis)	$ \mathcal{C}_i ^{-1} \sum_{v \in \mathcal{C}_i}  \mathcal{V} _z$
Top	$\max_{v \in \mathcal{C}_i}  \mathcal{V} _z$
Height	$\max_{v \in \mathcal{C}_i}  \mathcal{V} _z - \max_{v \in \mathcal{C}_i}  \mathcal{V} _z$
Bounding box	$\max \left\{ \max_{v \in \mathcal{C}_i}  \mathcal{V} _x - \max_{v \in \mathcal{C}_i}  \mathcal{V} _{x'}, \max_{v \in \mathcal{C}_i}  \mathcal{V} _y - \max_{v \in \mathcal{C}_i}  \mathcal{V} _y \right\}$
	$\min \left\{ \max_{v \in \mathcal{C}_i}  \mathcal{V} _x - \max_{v \in \mathcal{C}_i}  \mathcal{V} _{x'}, \max_{v \in \mathcal{C}_i}  \mathcal{V} _y - \max_{v \in \mathcal{C}_i}  \mathcal{V} _y \right\}$





**Figure 3** Normalized histograms of the variables conforming the feature vector employed by the person/non-person classifier.

computed. Observing Figure 3, we see that some variables are easily separable, i.e., weight, height, and bounding box. Moreover, they show a low cross dependency with other features.

A number of standard binary classifiers has been tested and their performances have been evaluated, namely Gaussian, Mixture of Gaussians, Neural Networks, K-Means, PCA, Parzen and Decision Trees [33,34]. Due to the aforementioned properties of the statistic distributions of the features, some classifiers are unable to obtain a good performance, i.e., Gaussian, PCA, etc. Other classifiers require a large number of characterizing elements, such as K-Means, MoG, or Parzen. Decision trees [33] have reported the best results. Separable variables such as height, weight, and bounding box size are automatically selected to build up a decision tree that yields a high recognition rate with a precision of 0.98 and a recall of 0.99 in our test database.

Another complementary criterium employed in the initiation of new tracks is based on the current state of the tracker. It will not be allowed to create a new track if its distance to the closest target is below a threshold.

### 2.3.2 Track termination criteria

A target will be deleted if one of the following conditions is fulfilled:

- If two or more tracks fall too close to one another, this indicates that they might be tracking the same target, hence only one will be kept alive while the rest will be removed.
- If tracker's efficiency becomes very low it might indicate that the target has disappeared and should be removed.

- The person/no-person classifier is applied to the set of features extracted from the voxels assigned to a target. If the classifier outputs a no-person verdict for a number of frames, the target will be considered as lost.

## 3 Voxel-based solutions

The filtering block shown in Figure 1 addresses the problem of keeping consistent trajectories of the tracked objects, resolving crossings among targets, mergings with spurious objects (i.e., shadows) and producing an accurate estimation of the centroid of the target based on the input voxel information. Although there is a number of papers addressing the problem of multi-camera/multi-person tracking, very few contributions have been based on voxel analysis [20,21].

### 3.1 PF tracking

PF is an approximation technique for estimation problems where the variables involved do not hold Gaussianity uncertainty models and linear dynamics. The current tracking scenario can be tackled by means of this algorithm to estimate the 3D position of a person  $\mathbf{x}_t = (x, y, z)_t$  at time  $t$ , taking as observation a set of colored voxels representing the 3D scene up to time  $t$  denoted as  $\mathbf{z}_{1:t}$ . For a given target  $\mathbf{x}_t$ , PF approximates the posterior density  $p(\mathbf{x}_t | \mathbf{z}_{1:t})$  as a sum of  $N_p$  Dirac functions:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \approx \sum_{j=1}^{N_p} w_t^j \delta(\mathbf{x}_t - \mathbf{x}_t^j), \quad (3)$$

where  $w_t^j$  are the weights associated to the particles, fulfilling  $\sum_j w_t^j = 1$ , and  $\mathbf{x}_t^j$  their positions. For this type of tracking problem, a sampling importance re-sampling (SIR) PF is applied to drive particles along time [16]. Assuming importance density to be equal to the prior density, weight update is recursively computed as

$$w_t^j \propto w_{t-1}^j p(\mathbf{z}_t | \mathbf{x}_t^j). \quad (4)$$

SIR PF avoids the particle degeneracy problem by re-sampling at every time step. In this case, weights are set to  $w_{t-1}^j = N_p^{-1}$ ,  $\forall j$ ; therefore,

$$w_t^j \propto p(\mathbf{z}_t | \mathbf{x}_t^j). \quad (5)$$

Hence, the weights are proportional to the likelihood function that will be computed over the incoming volume  $\mathbf{z}_t$ .

Finally, the best state at time  $t$ ,  $\tilde{\mathbf{x}}_t$ , is derived based on the discrete approximation of Equation 3. The most common solution is the Monte Carlo approximation of the expectation as

$$\tilde{\mathbf{x}}_t = E[\mathbf{x}_t | \mathbf{z}_{1:t}] \approx \sum_{j=1}^{N_p} w_t^j \mathbf{x}_t^j. \quad (6)$$

Basically, in the PF operation loop two steps must be defined: likelihood evaluation and particles propagation. In the following, we present our proposal for the PF implementation.

### 3.1.1 Likelihood evaluation

Binary and color information contained in  $\mathbf{z}_t$  will be employed to define the likelihood function  $p(\mathbf{z}_t | \mathbf{x}_t^j)$  relating the observation  $\mathbf{z}_t$  with the human body instance given by particle  $\mathbf{x}_t^j$ ,  $1 \leq j \leq N_p$ . Two partial likelihood functions,  $p_{\text{Raw}}(\mathbf{v}_t | \mathbf{x}_t^j)$  and  $p_{\text{Color}}(\mathbf{v}_t^C | \mathbf{x}_t^j)$ , will be combined linearly to produce  $p(\mathbf{z}_t | \mathbf{x}_t^j)$  as:

$$p(\mathbf{z}_t | \mathbf{x}_t^j) = \lambda p_{\text{Raw}}(\mathbf{v}_t | \mathbf{x}_t^j) + (1 - \lambda) p_{\text{Color}}(\mathbf{v}_t^C | \mathbf{x}_t^j). \quad (7)$$

Factor  $\lambda$  controls the influence of each term (foreground and color information) in the overall likelihood function. Empirical tests have shown that  $\lambda = 0.8$  provides satisfactory results. A more detailed review of the impact of color information in the overall performance of the algorithm is addressed in Section 5.1.

Likelihood associated to raw data is defined as the ratio of overlap between the input data  $\mathbf{v}_t$  and the ellipsoid  $\mathcal{E}_t^j$  defined by particle  $\mathbf{x}_t^j$  (see Section 2.3.1)

as

$$p_{\text{Raw}}(\mathbf{v}_t | \mathbf{x}_t^j) = \frac{|\mathbf{v}_t \cap \mathcal{E}_t^j|}{|\mathcal{E}_t^j|}. \quad (8)$$

For a given target  $k$ , an adaptive reference histogram  $\mathbf{H}_t^k$  of the colored surface voxels is available. This histogram is constructed using the YCbCr color space due to its robustness against light variations. The number of bins per channel will drive the ability of the system to distinguish between different color blobs; for our experiments, 21 bins per channel have been set empirically. The color likelihood function is constructed as

$$p_{\text{Color}}(\mathbf{v}_t^C | \mathbf{x}_t^j) = B(\mathbf{H}_t^k, H(\mathbf{v}_t^C \cap \mathcal{E}_t^j)), \quad (9)$$

where  $B(\cdot)$  is the Bhattacharya distance and  $H(\cdot)$  stands for the color histogram extraction operation of the enclosed volume. Update of the reference histogram is performed in a linear manner following the rule:

$$\mathbf{H}_t^k = \alpha \mathbf{H}_{t-1}^k + (1 - \alpha) H(\mathbf{v}_t^C \cap \mathcal{E}_t^{\tilde{\mathbf{x}}}), \quad (10)$$

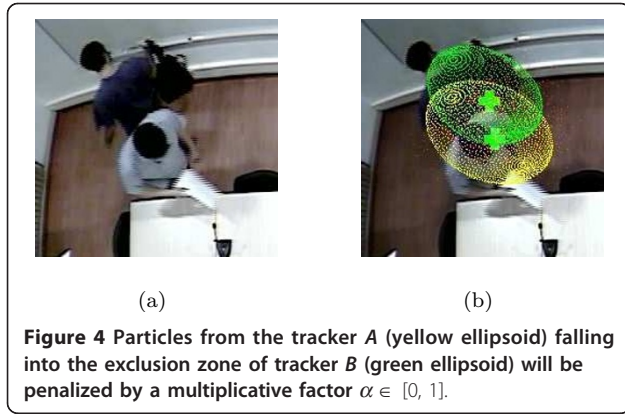
where  $\mathcal{E}_t^{\tilde{\mathbf{x}}}$  stands for the ellipsoid placed in the centroid estimation  $\tilde{\mathbf{x}}_t$  and  $\alpha$  is the adaptation coefficient. In our experiments,  $\alpha = 0.9$  provided satisfactory results.

### 3.1.2 Particle propagation

The propagation model has been chosen to be a Gaussian noise added to the state of the particles after the re-sampling step:  $\mathbf{x}_{t+1}^j = \mathbf{x}_t^j + \mathbf{N}$ . The covariance matrix  $\mathbf{P}$  corresponding to  $\mathbf{N}$  is proportional to the maximum variation of the centroid of the target and this information is obtained from the development part of the testing dataset. More sophisticated schemes employ previously learnt motion priors to drive the particles more efficiently [6]. However, this would penalize the efficiency of the system when tracking unmodeled motions patterns and, since our algorithm is intended for any motion tracking, no dynamical model is adopted.

### 3.1.3 Interaction model

Let us assume that there are  $N_T$  independent tracked targets. However, they are not fully independent since each tracker can consider voxels from other targets in both the likelihood evaluation and the 3D re-sampling step, resulting in target merging or identity mismatches. In order to achieve the most independent set of trackers, a blocking method to model interactions is considered. Some blocking proposals can be found in 2D tracking related studies [6] and an extension to the 3D domain is proposed. Blocking methods rely on penalizing particles whose associated ellipsoid model overlaps with other targets' ellipsoid as shown in Figure 4.



Hence, blocking information can also be considered when computing the particle weights for the  $k$ th target as

$$w_t^{k,j} = p(\mathbf{z}_t | \mathbf{x}_t^{k,j}) \prod_{\substack{l=1 \\ l \neq k}}^{N_T} \phi(\tilde{\mathbf{x}}_{t-1}^k, \tilde{\mathbf{x}}_{t-1}^l), \quad (11)$$

where  $\tilde{\mathbf{x}}_{t-1}^k$  stands for the estimation of the PF at time  $t - 1$  for target  $k$  and  $\phi(\cdot)$  is the blocking function defining exclusion zones that penalize particles from target  $l$  falling into the exclusion zone of target  $k$ . In this particular case, considering that people in the room are always sitting or standing up, this zone can be constrained to the  $xy$  plane. The proposed function is

$$\phi(\tilde{\mathbf{x}}_{t-1}^k, \tilde{\mathbf{x}}_{t-1}^l) = 1 - \exp\left(-k \left\| \begin{bmatrix} \tilde{\mathbf{x}}_{t-1}^k \\ \tilde{\mathbf{x}}_{t-1}^l \end{bmatrix}_{x,y} - \begin{bmatrix} \tilde{\mathbf{x}}_{t-1}^l \\ \tilde{\mathbf{x}}_{t-1}^k \end{bmatrix}_{x,y} \right\|^2\right), \quad (12)$$

where  $k \propto s_x^{-2}$  is the parameter that drives the sensibility of the exclusion zone.

### 3.2 SS tracking

In the presented PF tracking algorithm, likelihood evaluation can be computationally expensive, thus rendering this approach unsuitable for real-time systems. Moreover, data are usually noisy and may contain merged blobs corresponding to different targets. A new technique, SS, is proposed as an efficient and flexible alternative to PF.

Assuming a homogeneous 3D object, it can be proved that its centroid can exactly be computed based only on the surface voxels, since the interior voxels do not provide any relevant information. Hence, this centroid can be estimated through a discrete version of Green's theorem on the surface voxels [35,36], while other approaches obtain an accurate approximation of the centroid using feature points (see [37] for a review). A common assumption of these techniques is the

availability of surface data extracted beforehand, hence a labeling of the voxels in the scene should be available. By assuming that the object under study presents a central symmetry in the  $xy$  plane, the computation of the centroid can be done as an average of the positions of the surface voxels:

$$\tilde{\mathbf{x}}_t = \frac{\sum_{\mathcal{V} \in \mathcal{V}_t} [\mathcal{V}]_x}{|\mathcal{V}_t|} = \frac{\sum_{\mathcal{V} \in \mathcal{V}_t^s} [\mathcal{V}]_x}{|\mathcal{V}_t^s|}. \quad (13)$$

#### 3.2.1 Degree of mass and degree of surfaceness

Let us model the human body as an ellipsoid as previously done in the PF approach. In order to test the robustness of the centroid computation of Equation 13 against missing data, we studied the error committed when only a fraction of these input data is employed. A number of voxels (surface or interior voxels in each case) is randomly selected and employed to compute the centroid. Then, the error is computed showing that the surface-based estimation is more sensitive than the estimation using interior voxels (see Figure 5). However, this proves that the centroid can be computed from a number of randomly selected surface voxels still achieving a satisfactory performance. This idea is the underlying principle of the SS algorithm.

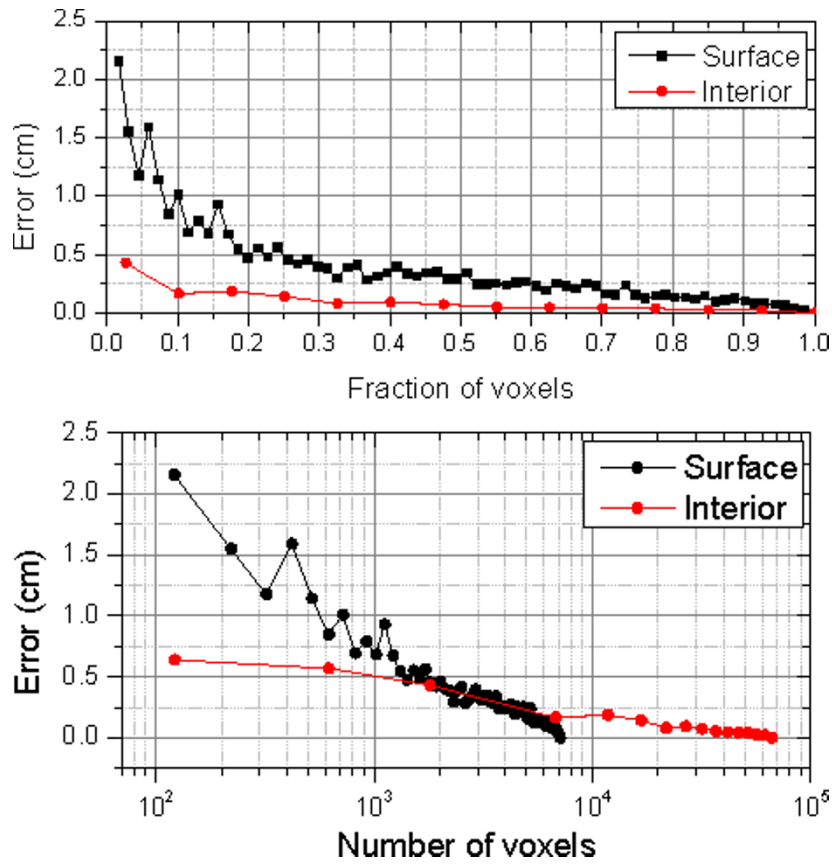
Let us estimate the centroid of an object by analyzing a randomly selected number of voxels from the whole scene, denoted as  $\mathcal{W}$ . An approach to the computation of the centroid would be

$$\tilde{\mathbf{x}}_t \approx \frac{\sum_{\mathcal{W} \in \mathcal{W}_t} \rho(\mathcal{W}) [\mathcal{W}]_x}{\sum_{\mathcal{W} \in \mathcal{W}_t} \rho(\mathcal{W})}, \quad \rho(\mathcal{W}) = \begin{cases} 1 & \text{if } \mathcal{W} \in \mathcal{V}_t \\ 0 & \text{if } \mathcal{W} \notin \mathcal{V}_t \end{cases} \quad (14)$$

where  $\rho(\mathcal{W})$  gives the mass density of voxel  $\mathcal{W}$ . Since it is assumed that all voxels have the same mass, this is a binary function that checks the occupancy of a given voxel. Hence, only the fraction of (randomly selected) voxels inside the object will contribute to the computation of the centroid. Equation 14 can be rewritten as

$$\tilde{\mathbf{x}}_t \approx \sum_{\mathcal{W} \in \mathcal{W}_t} \frac{\rho(\mathcal{W})}{\sum_{\mathcal{W} \in \mathcal{W}_t} \rho(\mathcal{W})} [\mathcal{W}]_x = \sum_{\mathcal{W} \in \mathcal{W}_t} \tilde{\rho}(\mathcal{W}) [\mathcal{W}]_x, \quad (15)$$

where  $\tilde{\rho}(\mathcal{W})$  can be considered as the normalized mass contribution of voxel  $\mathcal{W}$  to the computation of the centroid. If function  $\rho(\mathcal{W})$  takes values in the range [0,1] we may consider it as the "degree of mass" of  $\mathcal{W}$  or the importance of voxel  $\mathcal{W}$  into the calculation of  $\tilde{\mathbf{x}}_t$ . Then,  $\rho(\mathcal{W})$  might be considered as a normalized weight assigned to  $\mathcal{W}$ . Since we stated that the centroid



**Figure 5** Centroid's estimation error when computed with a fraction of surface or interior voxels. The employed ellipsoid had a radii  $s = (30, 30, 100)$  cm, and voxels with  $sv = 2$  cm were used.

can be computed using surface voxels, Equation 13 can be also posed as

$$\tilde{\mathbf{x}}_t \approx \sum_{\mathcal{W} \in \mathcal{W}_t} \frac{\rho_s(\mathcal{W})}{\sum_{\mathcal{W} \in \mathcal{W}_t} \rho(\mathcal{W})} [\mathcal{W}]_x = \sum_{\mathcal{W} \in \mathcal{W}_t} \tilde{\rho}_s(\mathcal{W}) [\mathcal{W}]_x, \quad (16)$$

where  $\rho_s(\mathcal{W}) \in [0, 1]$  measures the “degree of surface-ness” of voxel  $\mathcal{W}$ . Within this context, functions  $\rho(\cdot)$  and  $\rho_s(\cdot)$  might be understood as pseudo-likelihood functions and Equations 16 and 15 as a sample-based representation of an estimation problem.

### 3.2.2 Difference with particle filters

There is an obvious similarity between these representation and the formulation of particle filters but there is a significant difference. While particles in PF represent an instance of the whole body, our samples ( $\mathcal{W} \in \mathcal{W}_t$ ) are points in the 3D space. Moreover, particle likelihoods are computed over all data while sample pseudo-likelihoods will be computed in a local domain.

The presented concepts are applied to define the SS algorithm. Let  $\mathbf{y}_t^i \in \mathbb{R}^3$ , a point in the 3D space and  $\omega_t^i \in \mathbb{R}$  its associated weight measuring the pseudo-

likelihood of this position being part of the object or part of its surface. Under certain assumptions, it is achieved that the centroid can be computed as

$$\tilde{\mathbf{x}}_t \approx \sum_{i=1}^{N_s} \omega_t^i \mathbf{y}_t^i, \quad (17)$$

where  $N_s$  is the number of sampling points. When using SS we are no longer sampling the state space since  $\mathbf{y}_t^i$  cannot be considered an instance of the centroid of the target as happened with particles,  $\mathbf{x}_t^j$ , in PF. Hence, we will talk about *samples* instead of *particles* and we will refer to  $\{(\mathbf{y}_t^i, \omega_t^i)\}_{i=1}^{N_s}$  as the sampling set. This set will approximate the surface of the  $k$ th target,  $\mathcal{V}^{S,k}$ , and will fulfill the sparsity condition  $N_s \ll |\mathcal{V}^{S,k}|$ .

## 4 SS implementation

In order to define a method to recursively estimate  $\tilde{\mathbf{x}}_t$  from the sampling set  $\{(\mathbf{y}_t^i, \omega_t^i)\}_{i=1}^{N_s}$ , a filtering strategy has to be set. Essentially, the proposal is to follow the PF analysis loop (re-sampling, propagation, evaluation,



and estimation) with some opportune modifications to ensure the convergence of the algorithm.

#### 4.1 Pseudo-likelihood evaluation

Associated weight  $u_t^i$  to a sample  $y_t^i$  will measure the likelihood of that 3D position to be part of the surface of the tracked target. When computing the pseudo-likelihood, surface has been chosen instead of interior voxels, based on the efficiency of surface samples to propagate rapidly as will be explained in the next section. As in the defined PF likelihood function, two partial pseudo-likelihood functions<sup>b</sup>,  $p_{\text{Raw}}(\mathcal{V}_t | y_t^i)$  and  $p_{\text{Color}}(\mathcal{V}_t^C | y_t^i)$ , are linearly combined to form  $p(z_t | y_t^i)$  as

$$p(z_t | y_t^i) = \lambda p_{\text{Raw}}(\mathcal{V}_t | y_t^i) + (1 - \lambda) p_{\text{Color}}(\mathcal{V}_t^C | y_t^i). \quad (18)$$

Partial likelihoods will be computed on a local domain centered in the position  $y_t^i$ . Let  $\mathcal{C}(y_t^i, q, r)$  be a neighborhood of radius  $r$  over a connectivity  $q$  domain on the 3D orthogonal grid around a sample place in a voxel position  $y_t^i$ . Then, we define the occupancy and color neighborhoods around  $y_t^i$  as  $\mathcal{O}_t^i = \mathcal{V}_t \cap \mathcal{C}(y_t^i, q, r)$  and  $\mathcal{C}_t^i = \mathcal{V}_t^C \cap \mathcal{C}(y_t^i, q, r)$ , respectively.

For a given sample  $y_t^i$  occupying a single voxel, its weight associated to the raw data will measure its likelihood to belong to the surface of an object. It can be modeled as

$$p_{\text{Raw}}(\mathcal{V}_t | y_t^i) = 1 - \left| \frac{2 |\mathcal{O}_t^i|}{|\mathcal{C}(y_t^i, q, r)|} - 1 \right|. \quad (19)$$

Ideally, when the sample  $y_t^i$  is placed in a surface, half of its associated occupancy neighborhood will be occupied and the other half empty. The proposed expression attains its maximum when this condition is fulfilled.

Function  $p_{\text{Color}}(\mathcal{V}_t^C | y_t^i)$  can be defined as the likelihood of a sample belonging to the surface corresponding to the  $k$ th target characterized by an adaptive reference color histogram  $\mathbf{H}_t^k$ :

$$p_{\text{Color}}(\mathcal{V}_t^C | y_t^i) = D(\mathbf{H}_t^k, \mathcal{C}_t^i). \quad (20)$$

Since  $\mathcal{C}_t^i$  contains only local color information with reference of the global histogram  $\mathbf{H}_t^k$ , the distance  $D(\cdot)$  is constructed toward giving a measure of the likelihood between this local colored region and  $\mathbf{H}_t^k$ . For every voxel in  $\mathcal{C}_t^i$ , it is decided whether it is similar to  $\mathbf{H}_t^k$  by selecting the histogram value for the tested color and checking whether it is above a threshold  $\gamma$  or not. Finally, the ratio between the number of similar color and total voxels in the neighborhood gives the color

similarity score. Since reference histogram is updated and changes over time, a variable threshold  $\gamma$  is computed, so that the 80% of the values of  $\mathbf{H}_t^m$  are taken into account.

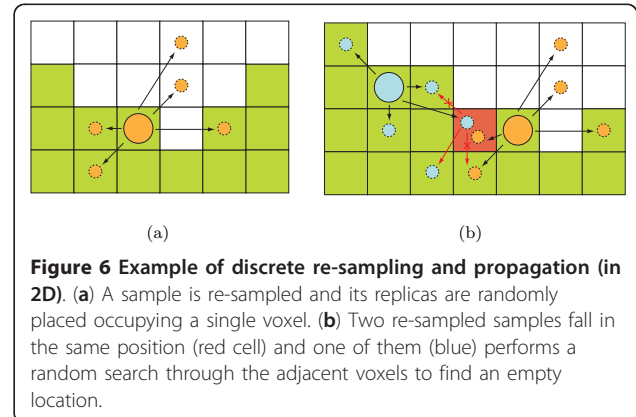
One of the advantages of the SS algorithm is its computational efficiency. The complexity to compute  $p(z_t | y_t^i)$  is quite reduced since it only evaluates a local neighborhood around the sample in comparison with the computational load required to evaluate the likelihood of a particle in the PF algorithm. This point will be quantitatively addressed in Section 5.2.

The parameters defining the neighborhood were set to  $q = 26$  and  $r = 2$  yielding to satisfactory results. Larger values of the radius  $r$  did not significantly improve the overall algorithm performance but increased its computational complexity.

#### 4.2 Sample propagation and 3D discrete resampling

A sample  $y_t^i$  placed near a surface will have an associated weight  $\omega_t^i$  with a high value. It is a valid assumption to consider that some surrounding positions might also be part of this surface. Hence, placing a number of new particles in the vicinity of  $x_t^j$  would contribute to progressively explore the surface of a voxel set. This idea leads to the spatial re-sampling and propagation scheme that will drive samples along time in the surface of the tracked target.

Given the discrete nature of the 3D voxel space, it will be assumed that every sample is constrained to occupy a single voxel or discrete 3D coordinate and there cannot be two samples placed in the same location. Re-sampling is mimicked from PF so a number of replicas proportional to the normalized weight of the sample are generated. Then, these new samples are propagated and some *discrete* noise is added to their position meaning that their new positions are also constrained to occupy a discrete 3D coordinate (see an example in Figure 6). However, two re-sampled and propagated particles may



**Figure 6 Example of discrete re-sampling and propagation (in 2D).** (a) A sample is re-sampled and its replicas are randomly placed occupying a single voxel. (b) Two re-sampled samples fall in the same position (red cell) and one of them (blue) performs a random search through the adjacent voxels to find an empty location.

fall in the same 3D voxel location as shown in Figure 6. In such case, one of these particles will randomly explore the adjacent voxels until reaching an empty location; if there is not any suitable location for this particle, it will be dismissed.

The choice of sampling the surface voxels of the object instead of its interior voxels to finally obtain its centroid is motivated by the fact that propagating samples along the surface rapidly spread them all around the object as depicted in Figure 7. Propagating samples on the surface is equivalent to propagate them on a 2D domain, hence the condition of not placing two samples in the same voxel will make them to explore the surface faster (see Figure 6). On the other hand, interior voxels propagate on a 3D domain, thus having more space to explore and therefore becoming slower to spread all around the volume (see Figure 6). Although both (pseudo-)likelihoods should produce a fair estimation of the object's centroid, both sampling sets must fulfill the condition to be randomly spread around the object volume, otherwise the centroid estimation will be biased.

#### 4.2.1 Interaction model

The flexibility of a sample-based analysis may, sometimes, lead to situations where particles spread out too much from the computed centroid. In order to cope with this problem, an intra-target samples' interaction model is devised. If a sample is placed in a position such that  $\| [y_t^i]_{x,y} - [\tilde{x}_{t-1}]_{x,y} \| > \delta$  it will be removed (that is to assign  $\omega_t^i = 0$ ) and we set the threshold as  $\delta = \alpha s_x$ , with  $s_x = 30$  cm. Factor  $\alpha = 1.5$  produced accurate results in our experiments.

The interaction among targets is modeled in similar way as in the PF approach. Formulas in Equations 11 and 12 are applied to samples with the appropriate scaling parameter  $k$ .

## 5 Results and evaluation

In order to assess the performance of the proposed tracking systems, they have been tested on the set of

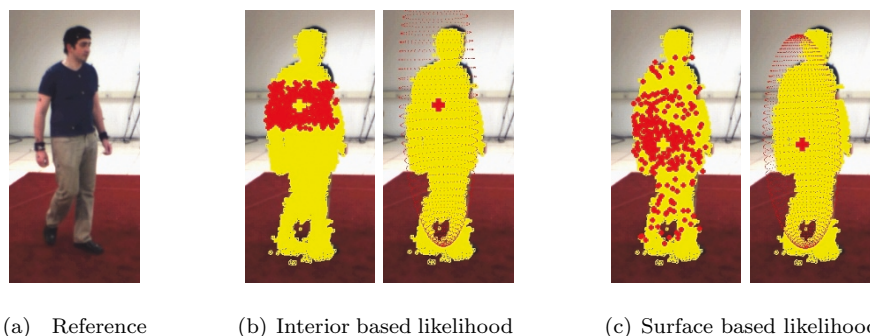
benchmarking image sequences provided by the CLEAR Evaluation Campaigns 2007 [22]. Typically, these evaluation sequences involved up to five people moving around in a meeting room. This benchmarking set was formed by two separate datasets, development, and evaluation, containing sequences recorded by five of the participating partners. A sample of these data can be seen in Figure 8. The development set consisted in 5 sequences of an approximate duration of 20 min each, while the evaluation set was formed by 40 sequences of 5 min each, thus adding up to 5 h of data. Each sequence was recorded with four cameras placed in the corners of the room and a zenithal camera placed in the ceiling. All cameras were calibrated and had resolutions ranging from  $640 \times 480$  to  $756 \times 576$  pixels at an average frame rate of  $f_R = 25$  fps. The test environments were a  $5 \times 4$  m rooms with occluding elements such as tables and chairs. Images of the empty rooms were also provided to train the background/foreground segmentation algorithms.

Metrics proposed in [4] for multi-person tracking evaluation have been adopted, namely the Multiple Object Tracking Precision (*MOTP*), which shows tracker's ability to estimate precise object positions, and the Multiple Object Tracking Accuracy (*MOTA*), which expresses its performance at estimating the number of objects, and at keeping consistent trajectories. *MOTP* scores the average metric error when estimating multiple target 3D centroids, while *MOTA* evaluates the percentage of frames where targets have been missed, wrongly detected or mismatched.

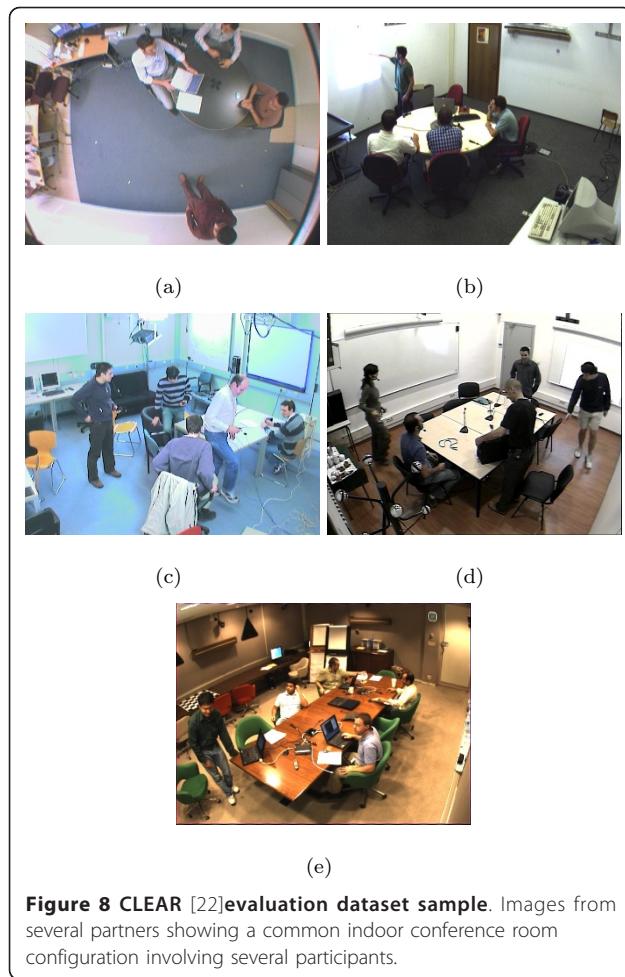
The aim of a tracking system would be to produce high values of *MOTA* and low values of *MOTP* thus indicating its ability to correctly track all targets and estimate their positions accurately. When comparing two algorithms, there will be a preference to choose the one outputting the highest *MOTA* score.

### 5.1 Results

To demonstrate the effectiveness of the proposed multi-person tracking approaches, a set of experiments were



**Figure 7** Sample positions evolution and centroid estimation. Likelihood based on: (a) interior voxels, or (b) surface voxels.



**Figure 8** CLEAR [22] evaluation dataset sample. Images from several partners showing a common indoor conference room configuration involving several participants.

conducted over the CLEAR 2007 database. The development part of the dataset was used to train the initiation/termination of tracks modules as described in Section 2.3 and the remaining test part was used for our experiments.

First, the multi-camera data are pre-processed performing the foreground and background segmentations and 3D voxel reconstruction algorithm. In order to analyze the dependency of the tracker's performance with the resolution of the 3D reconstruction, several voxel sizes were employed  $s_v = \{2, 5, 10, 15\}$  cm. A colored version of these voxel reconstructions was also generated, according to the technique introduced in Section 2.1. Then, these data were the input fed to the PF and SS proposed approaches.

In both types of filters, SS or PF, three parameters drive the performance of the algorithm: the voxel size  $s_v$ , the number of samples  $N_s$ , or particles  $N_p$ , and the usage of color information. Experiments carried out explore the influence of these parameters in the MOTP, precision in cm., and MOTA, tracker accuracy (in % of

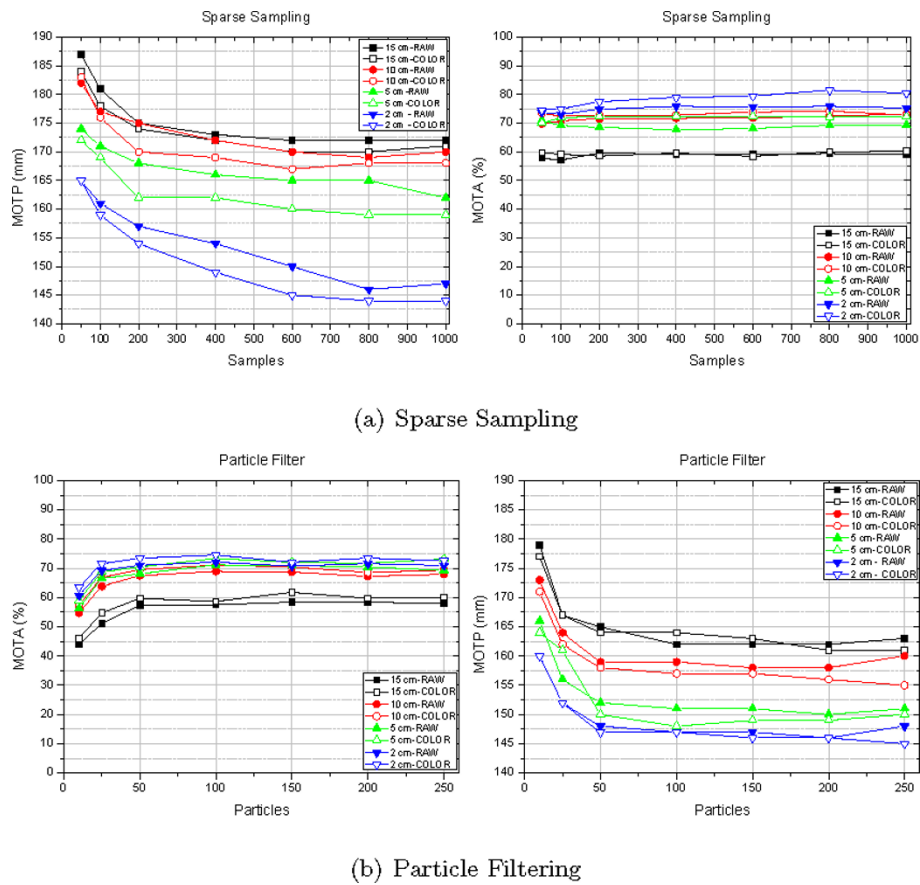
correctly tracked targets), shown in Figure 9. Some remarks can be drawn

- **Number of samples/particles:** There is a dependency between the MOTP score and the number of particles/samples, especially for the SS algorithm. The contribution of a new sample to the estimation of the centroid in the SS has less impact than the addition of a new particle in the PF, hence the slower decay of the MOTP curves for the SS than for the PF. Regarding the MOTA score, there is not a significant dependency with  $N_s$  or  $N_p$ . Two factors drive the MOTA of an algorithm: the track initiation/termination modules, that mainly contributes to the ratio of misses and false positives, and the filtering step that has an impact to the mismatches ratio. The low dependency of MOTA with  $N_s$  or  $N_p$  shows that most of the impact of the algorithm in this score is due to the particle/sample propagation and interaction strategies rather than the quantity of particles/samples itself. Moreover, the influence in the MOTA score is tightly correlated with the track initiation/termination policy. This assumption was experimentally validated by testing several classification methods (mixture of Gaussians, PCA, Parzen, and K-Means) in the initiation/termination modules yielding to a drop in the MOTA score proportional to their ability to correctly classify a blob as person/no-person.

- **Voxel size:** Scenes reconstructed with a large voxel size do not capture well all spatial details and may miss some objects thus decreasing the performance of the system (both in SS and PF). It can be observed that MOTP and MOTA scores improve as the voxel size decrease.

- **Color features:** Color information improves the performance of SS and PF in both MOTP and MOTA scores. First, there is an improvement when using color information for a given voxel size, specially for the SS algorithm. Moreover, the smaller the voxel size the most noticeable difference between the experiments using raw and color features. This effect is supported by the fact that color characteristics are better captured when using small voxel sizes. The performance improvement when using color in the SS algorithm is more noticeable since samples are placed in the regions with a high likelihood to be part of the target. For instance, this effect is more evident in cases where the subject is sitting and the particles concentrate in the upper body part, disregarding the part of the chair. In the SS algorithm, MOTP score benefits from this efficient sample placement. PF algorithm is constrained to evaluate the color likelihood in the ellipsoid defined in Equation 9 thus not being able to differentiate between parts of the blob that do not belong to the tracked target. Color information used within the filtering loop leads to a better distinguishability among blobs, thus reducing the mismatch ratio and



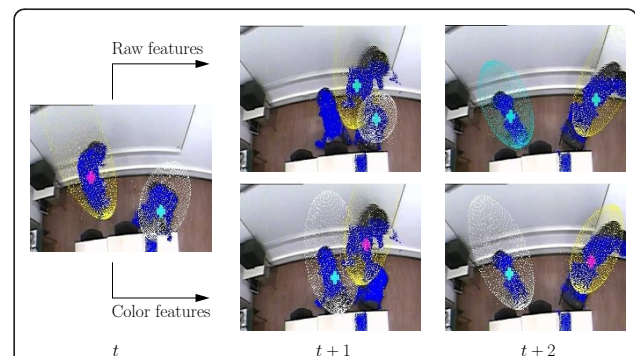


**Figure 9** MOTP and MOTA scores for the SS and the PF techniques using raw and colored voxels. Several voxel sizes  $sv = \{2, 5, 10, 15\}$  cm have been used in the experiments.

slightly improving the MOTA score. Merging of adjacent blobs or complex crossing among targets is also correctly resolved. An example of the impact of color information is shown in Figure 10 where the usage of color avoids the mismatch between two targets. This effect is more noticeable when targets in the scene are dressed in different colors.

We can compare the results obtained by SS and PF with other algorithms evaluated using the same CLEAR 2007 database whose scores are reported in Table 2. Most of these methods exploited multi-view information with the exception of [31] that only used the zenithal camera facing the associated distortion and perspective problems. PF is the most employed technique due to its suitability to the characteristics of this problem although Kalman filtering used by [15] provided fair results when fed by higher semantical features extracted from the input data (in this case, faces). Note the low *FP* score for this system as a consequence of the unlikely event of detecting a face in a spurious object. A 3D voxel reconstruction was used as the input data in [5] together with a simple track management system. The rest of the

methods [7,31] relied on a fixed human body appearance model similar to the ellipsoidal region of interest used in our PF proposal. However, the novelty of these



**Figure 10** Zenithal view of two comparative experiments showing the influence of color in the SS algorithm. The cross-over between two targets is correctly tackled when using color information whereas using only raw features leads to a mismatch and, afterwards, a track loss (white ellipsoid) and the initiation of a new one (cyan ellipsoid).



**Table 2 Results presented at the CLEAR 2007 [22] by several partners**

Method	MOTP (mm)	MOTA (%)	FP (%)	Miss (%)	MM (cases)
Face detection+Kalman filtering [15]	91	59.66	06.99	30.89	2.46
Appearance models+PF [7]	141	59.62	18.58	20.66	1.14
Upper body detection+PF [31]	155	69.58	14.50	15.09	0.83
Zenithal camera analysis+PF [31]	222	54.94	20.24	23.74	1.08
Voxel analysis+Heuristic tracker [5]	168	30.49	40.19	27.74	1.58
Voxel analysis+PF (best case)	147	74.56	14.03	10.48	0.91
Voxel analysis+SS (best case)	144	81.50	09.34	08.70	0.46

Multi-camera information is used to track multiple people using several methods

methods is the strategies to combine the information coming from the analysis of different views without performing any 3D reconstruction. Comparing the best proposed tracking system [31]<sup>c</sup> with our two approaches, we obtain a relative improvement of  $\Delta(MOTP, MOTA)_{ss} = (7.63, 17.13)\%$  and  $\Delta(MOTP, MOTA)_{pf} = (5.16, 7.15)\%$ .

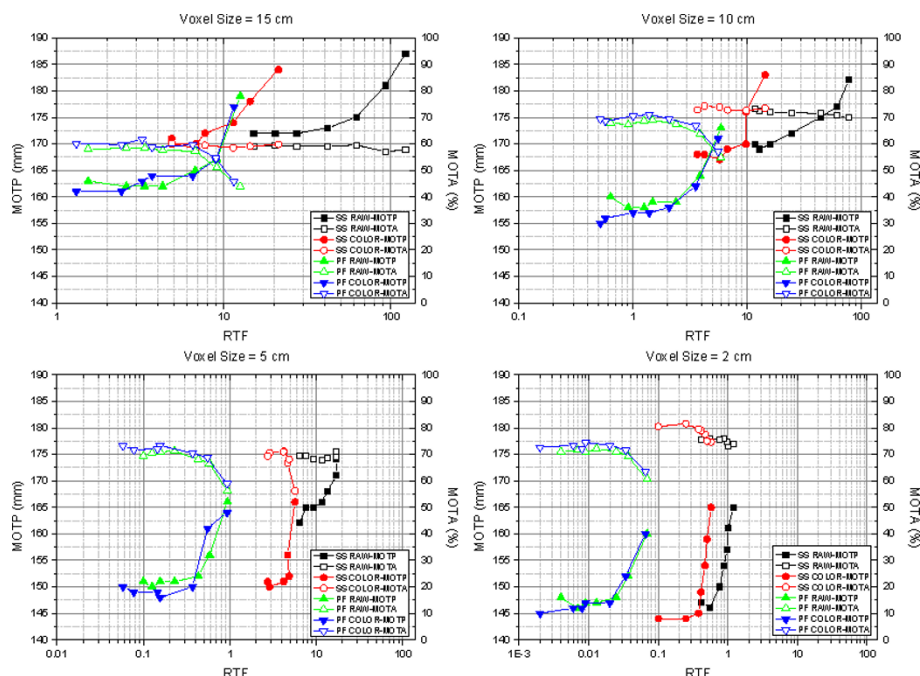
In order to visually show the performance of the SS algorithm, some videos corresponding to the most challenging tracking scenarios have been made available at <http://www.cristiancanton.org>.

## 5.2 Computational performance

Comparing obtained metrics among different algorithms can give an idea about their performance in a scenario where computational complexity is not taken into account. An analysis of the operation time of several algorithms under the same conditions and the produced MOTP/MOTA metrics might give a more informative and fairer comparison tool. Although there is not a standard procedure to measure the computational performance of a tracking process, we devised a method to assess the computational efficiency of our algorithms to present a comparative study.

The RTFfactor associated with a performance measure MOTP/MOTA (in both vertical axes) of the SS and PF algorithms when dealing with raw and colored input voxels is presented in Figure 11. This factor indicates a proportional measure of the speed of the algorithm where  $RTF = 1$  stands for real-time operation while  $RTF > 1$  and  $RTF < 1$  indicate a faster or slower performance, respectively. Each point of every curve is the result of an experiment conducted over all the CLEAR data set associated to a number of samples/particles of each algorithm.

The first noticeable characteristic of these charts is that, due to the computational complexity of each algorithm, when comparing SS and PF algorithms under the same operation conditions, the RTF associated with SS



**Figure 11 Computational performance comparison between PF and SS using several voxel sizes  $s_V = \{2, 5, 10, 15\}$  cm and features (raw or colored voxels). MOTP and MOTA scores are related to the real-time factor (RTF) showing the computational load required by each algorithm to attain a given tracking performance.**

is always higher than the associated with PF. Similarly, the computational load is higher when analyzing colored than raw inputs. All the plotted curves attain lower RTF performance values as the size of the voxel  $s_v$  decreases since the amount of data to process increases (note the different RTF scale ranges for each voxel size in Figure 11). Regarding the MOTP/MOTA metrics, there is a common tendency to a decrease in the MOTP and an increase in the MOTA as the RTF decreases. The separation between the SS and PF curves is bigger as the voxel size decreases since the PF algorithm has to evaluate a larger amount of data.

The observation of these results yields the conclusion that the SS algorithm is able to produce a similar and, in some cases, better results than the PF algorithm with a lower computational cost. For example, using  $s_v = 5$  cm, a MOTP score of around 165 mm can be obtained using SS with a RTF ten times larger than when using PF and similarly with the MOTA score.

## 6 Conclusions

In this article, we have presented a number of contributions to the multi-person tracking task in a multi-camera environment. A block representation of the whole tracking process allowed to identify the performance bottlenecks of the system and address efficient solutions to each of them. Real-time performance of the system was a major goal hence efficient tracking algorithms have been produced as well as an analysis of their performance.

The performance of these systems has thoroughly been tested over the CLEAR database and quantitatively compared through two scores: MOTP and MOTA. A number of experiments have been conducted toward exploring the influence of the resolution of the 3D reconstruction and the color information. Results have been compared with other state-of-the-art algorithms evaluated with the same metrics using the same testing data.

The relevance of the initiation and termination of filters have been proved, since these modules have a major impact on the MOTA score. However, most articles in the literature do not specifically address the operation of these modules. We proposed a statistical classifier based on classification trees as a way to discriminate blobs between the person/no-person classes. Training of this classifier was done using data available in the development part of the employed database and a number of features (namely weight, height, top in  $z$ -axis, bounding box size) were extracted and provided as the input to the classifier. Another criterium such as a proximity to other already existing tracks was employed to create or destroy a track. Performance scores in Table 2 for the PF and SS systems present the lowest

values for the false positives (*FP*) and missed targets (*Miss*) ratios hence supporting the relevance of the initiation and termination of tracks modules.

Two proposals for the filtering step of the tracking system have been presented: PF and SS. An independent tracker was assigned to every target and an interaction model was defined. PF technique proved to be robust and led to state-of-the-art results but its computational load was unaffordable for small voxel sizes. As an alternative, SS algorithm has been presented achieving a similar and, in some occasions, better performance than PF at a smaller computational cost. Its sample-based estimation of the centroid allowed a better adaptation to noisy data and distinguishability among merged blobs. In both PF and SS, color information provided a useful cue to increase the robustness of the system against track mismatches thus increasing the MOTA score. In the SS, color information also allowed a better placement of the samples allowing to distinguish among parts belonging to the tracked object and parts of a merging with a spurious object, leading to a better MOTP score.

Future research within this topic involves multi-modal data fusion with audio data toward improving the precision of the tracker, MOTP, and avoid mismatches among targets, thus improving the MOTA score.

## End notes

<sup>a</sup>Analogously to the pixel definition (*picture element*) as the minimum information unit in a discrete image, the voxel (*volume element*) is defined as the minimum information unit in a 3D discrete representation of a volume.

<sup>b</sup>For the sake of simplicity in the notation, pseudo-likelihood functions will be denoted as  $p(\cdot)$  instead of defining a specific notation for it.

<sup>c</sup>When selecting the best system, the MOTA score is regarded as the most significant value.

The authors declare that they have no competing interests

Received: 15 May 2011 Accepted: 23 November 2011  
Published: 23 November 2011

## References

1. S Park, MM Trivedi, Understanding human interactions with track and body synergies captured from multiple views. *Comput Vis Image Understand* 111(1), 2–20 (2008). doi:10.1016/j.cviu.2007.10.005
2. Project CHIL—Computers in the Human Interaction Loop, <http://chil.server.de> (2004–2007)
3. I Haritaoglu, D Harwood, LS Davis,  $W^4$ : real-time surveillance of people and their activities. *IEEE Trans Pattern Anal Mach Intell.* 22(8), 809–830 (2000). doi:10.1109/34.868683
4. K Bernardin, A Elbs, R Stiefelwagen, Multiple object tracking performance metrics and evaluation in a smart Room environment, in *Proceedings of IEEE International Workshop on Visual Surveillance* (2006)
5. C Canton-Ferrer, J Salvador, JR Casas, Multi-person tracking strategies based on voxel analysis, in *Proceedings of Classification of Events, Activities and*

- Relationships Evaluation and Workshop*, vol. 4625. Lecture Notes on Computer Science, 91–103 (2007)
6. Z Khan, T Balch, F Dellaert, Efficient particle filter-based tracking of multiple interacting targets using an MRF-based motion model, in *Proceedings of International Conference on Intelligent Robots and Systems*. **1**(1), 254–259 (2003)
  7. O Lanz, P Chippendale, R Brunelli, An appearance-based particle filter for visual tracking in smart rooms, in *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, vol. 4625. Lecture Notes on Computer Science, 57–69 (2007)
  8. A Yilmaz, O Javed, M Shah, Object tracking: a survey. *ACM Comput Surv.* **38**(4), 1–45 (2006)
  9. C Canton-Ferrer, JR Casas, M Pardàs, Towards a Bayesian approach to robust finding correspondences in multiple view geometry environments, in *Proceedings of 4th International Workshop on Computer Graphics and Geometric Modelling*, vol. 3515. Lecture Notes on Computer Science, 281–289 (2005)
  10. O Lanz, Approximate Bayesian multibody tracking. *IEEE Trans Pattern Anal Mach Intell.* **28**(9), 1436–1449 (2006)
  11. GKM Cheung, T Kanade, JY Bouguet, M Holler, A real time system for robust 3D voxel reconstruction of human motions, in *IEEE Conference on Computer Vision and Pattern Recognition* **2**, 714–720 (2000)
  12. J Isidoro, S Sclaroff, Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints, in *Proceedings of IEEE International Conference on Computer Vision* **2**, 1335–1342 (2003)
  13. I Mikić, S Santini, R Jain, Tracking objects in 3D using multiple camera views, in *Proceedings of Asian Conference on Computer Vision* (2000)
  14. D Focken, R Stiefelhagen, Towards vision-based 3-D people tracking in a Smart Room, in *Proceedings of IEEE International Conference on Multimodal Interfaces*, 400–405 (2002)
  15. N Katsarakis, F Talantzis, A Pnevmatikakis, L Polymenakos, The AIT 3D audiovisual person tracker for CLEAR 2007, in *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, vol. 4625. Lecture Notes on Computer Science, 35–46 (2007)
  16. MS Arulampalam, S Maskell, N Gordon, T Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process.* **50**(2), 174–188 (2002). doi:10.1109/78.978374
  17. K Lien, C Huang, Multiview-based cooperative tracking of multiple human objects. *EURASIP J. Image Video Process* **8**(2), 1–13 (2008)
  18. T Osawa, X Wu, K Sudo, K Wakabayashi, H Arai, MCMC based multi-body tracking using full 3D model of both target and environment, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, 224–229 (2007)
  19. J Black, T Ellis, P Rosin, Multi view image surveillance and tracking, in *Proceedings of Workshop on Motion and Video Computing*, 169–174 (2002)
  20. A López, C Canton-Ferrer, JR Casas, Multi-person 3D tracking with particle filters on voxels, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* **1**, 913–916 (2007)
  21. C Canton-Ferrer, R Sblendido, JR Casas, M Pardàs, Particle Filtering and sparse sampling for multi-person 3D tracking, in *Proceedings of IEEE International Conference on Image Processing*, 2644–2647 (2008)
  22. CLEAR—Classification of Events, Activities and Relationships Evaluation and Workshop, <http://www.clear-evaluation.org> (2007)
  23. DL Hall, SAH McMullen, *Mathematical Techniques in Multisense Data Fusion*. Artech House (2004)
  24. KN Kutulakos, SM Seitz, A theory of shape by space carving. *Int J Comput Vis.* **38**(3), 199–218 (2000). doi:10.1023/A:1008191222954
  25. O Faugeras, R Keriven, Variational principles, surface evolution, PDE's, level set methods and the stereo problem, in *Proceedings of 5nd IEEE EMBS International Summer School on Biomedical Imaging* (2002)
  26. JR Casas, J Salvador, Image-based multi-view scene analysis using conexels, in *Proceedings of HCSNet Workshop on Use of Vision in Human-Computer Interaction*, 19–28 (2006)
  27. V Kolmogorov, R Zabini, What energy functions can be minimized via graph cuts?. *IEEE Trans Pattern Anal Mach Intell.* **26**(2), 147–159 (2004). doi:10.1109/TPAMI.2004.1262177
  28. C Stauffer, W Grimson, Adaptive background mixture models for real-time tracking, in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 252–259 (1999)
  29. E Maggio, E Piccardo, C Regazzoni, A Cavallaro, Particle PHD filtering for multi-target visual tracking, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. **1**, 1101–1104 (2007)
  30. F Talantzis, A Pnevmatikakis, AG Constantinides, Audio-visual active speaker tracking in cluttered indoors environments. *IEEE Trans Syst Man Cybern B.* **38**(3), 799–807 (2008)
  31. K Bernardin, T Gehrig, R Stiefelhagen, Multi-level particle filter fusion of features and cues for audio-visual person tracking, in *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, vol. 4625. Lecture Notes on Computer Science, 70–81 (2007)
  32. JW Tuckey, *Exploratory Data Analysis*. Addison-Wesley (1977)
  33. L Breiman, JH Friedman, RA Olshen, CJ Stone, *Classification and Regression Trees*. Chapman and Hall (1993)
  34. RO Duda, PE Hart, DG Stork, *Pattern Classification*. Wiley-Interscience (2000)
  35. JJ Crisco, RD McGovern, Efficient calculation of mass moments of inertia for segmented homogeneous three-dimensional objects. *J Biomech.* **31**(1), 97–101 (1998)
  36. JG Leu, Computing a shape's moments from its boundary. *Pattern Recogn.* **24**(10), 116–122 (1991)
  37. L Yang, F Albrechtsen, Fast and exact computation of Cartesian geometric moments using discrete Green's theorem. *Pattern Recogn.* **29**(7), 1061–1073 (1996). doi:10.1016/0031-3203(95)00147-6

doi:10.1186/1687-6180-2011-114

**Cite this article as:** Canton-Ferrer et al.: Multi-camera multi-object voxel-based Monte Carlo 3D tracking strategies. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:114.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)