PARTICLE FILTERING AND SPARSE SAMPLING FOR MULTI-PERSON 3D TRACKING

C. Canton-Ferrer, J.R. Casas, M. Pardàs

Technical University of Catalonia
Barcelona, Spain

R. Sblendido

Politecnico di Milano
Milano, Italy

ABSTRACT

This paper 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. 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 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 annotated databases yield quantitative results showing the effectiveness of the proposed algorithms in indoor scenarios.

Index Terms— Multi-target tracking, particle filtering, 3D color processing, multi-camera analysis, human-computer interfaces

1. INTRODUCTION

The current paper addresses the problem of detecting and tracking a group of people present in an indoor scenario in a multiple camera setup. Robust, multi-person tracking systems are employed in a wide range of applications, including SmartRoom environments, surveillance for security, health monitoring, as well as providing location and context features for human-computer interaction.

A number of methods for camera based multi-person 3D tracking has been proposed in the literature [1, 2, 3, 4]. A common goal in these systems is robustness under occlusions created by multiple objects present in the scene when estimating the position of a target. Single camera approaches [2] have been widely employed but are more vulnerable to occlusions, rotation and scale changes of the target. In order to avoid these drawbacks, multi-camera tracking techniques [3] exploit spatial redundancy among different views and provide 3D information as well. Integration of features extracted from multiple cameras has been proposed in terms of image correspondences [5], multi-view histograms [4] or voxel reconstructions [6].

Filtering techniques are employed to add temporal consistency to tracks. Kalman filtering approaches have been extensively used to track a single object under Gaussian uncertainty models and linear dynamics [2]. However, these methods do not perform accurately when facing noisy scenes or rapidly maneuvering targets. Particle filtering has been applied to cope with these situations since it can deal with multi-modal *pdf*s and is able to recover from lost tracks [1, 7].

In this paper, we propose two methods for 3D tracking of multiple people in a multi-camera environment. Redundancy among cameras is exploited to obtain a colored 3D voxel representation of the scene as the input for the tracking systems. Our first proposal is to employ a particle filter to track a target estimating its 3D centroid, assuming a fixed size ellipsoid as the human body model of a person. Particle weights are evaluated through a volume likelihood function taking into account both occupancy and color information. Multiple targets are tracked assigning a particle filter to every one together with a color model in order to increase robustness against mismatches among them. In order to achieve the most independent set of trackers, we consider a 3D blocking method to model interactions. The second proposed method aims at decreasing computation time by means of a novel tracking technique based on the seminal particle filtering principle extending our previous research [8]. 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 algorithms is assessed by means of objective metrics defined in the framework the CLEAR [9] multi-target tracking database.

2. SYSTEM OVERVIEW

For a given frame in the video sequence, a set of N images are obtained from the N cameras (see a sample in Fig.1a). 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 substraction technique [10] as shown in Fig.1b.

Redundancy among cameras is exploited by means of a Shape-from-Silhouette (SfS) technique [6]. This process generates a discrete occupancy representation of the 3D space (voxels). A voxel is labelled as foreground or background by checking the spatial consistency of its projection on the N segmented silhouettes. The data obtained with this 3D reconstruction is corrupted by spurious voxels introduced due to wrong segmentation, camera calibration inaccuracies, etc. A connectivity filter is introduced in order to remove these voxels and the final 3D binary reconstruction is shown in Fig.1c. The visibility of a surface voxel onto a given camera is 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 voxels is assigned to it. An example of this process is depicted in Fig.1d.

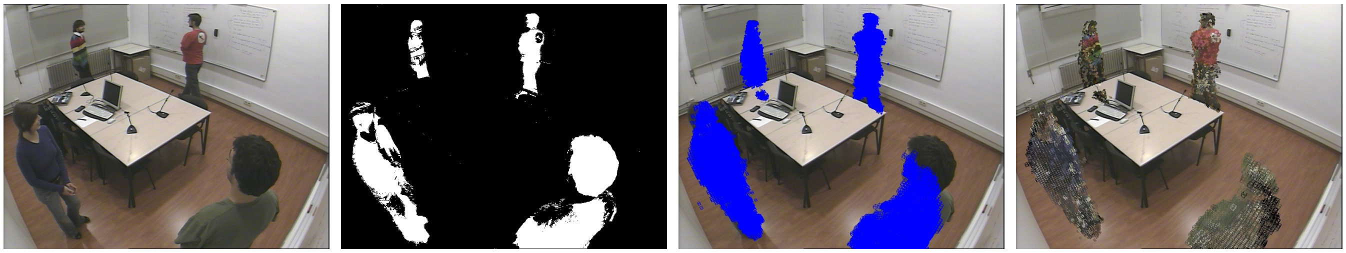
(a) (b) (c) (d)

Fig. 1. Input data generation example. In (a), a sample of the original images. In (b), foreground segmentation of the input images employed by the SfS algorithm. In (c), example of the binary 3D voxel reconstruction and, in (d), the final colored version shown over a background image.

The resulting colored 3D scene reconstruction is fed to the proposed systems that assign a tracker to each target. The resulting tracks are processed by a higher semantic analysis module. Information about the environment (dimensions of the room, furniture, etc.) allow discarding tracks that are clearly wrong.

3. PARTICLE FILTERING

Particle Filtering (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 $\mathbf{z}_{1:t}$. Multiple people might be tracked assigning a PF to each target and defining an interaction model to ensure track coherence.

For a given target $\mathbf{x}_t$, PF approximates the posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ with a sum of N_s Dirac functions:

$$p\left(\mathbf{x}_t|\mathbf{z}_{1:t}\right) \approx \sum_{j=1}^{N_s} w_t^j \delta(\mathbf{x}_t - \mathbf{x}_t^j), \tag{1}$$

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 [7]. Assuming importance density to be equal to the prior density, weight update is recursively computed as:

$$w_t^j \propto w_{t-1}^j \, p(\mathbf{z}_t|\mathbf{x}_t^j). \tag{2}$$

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

$$w_t^j \propto p(\mathbf{z}_t|\mathbf{x}_t^j). \tag{3}$$

Hence, the weights are proportional to the likelihood function that will be computed over the incoming volume $\mathbf{z}_t$. The re-sampling step derives the particles depending on the weights of the previous step, then all the new particles receive a starting weight equal to $1/N_s$ which will be updated by the next volume likelihood function.

Finally, the best state at time t of target m, $\mathbf{X}_t^m$, is derived based on the discrete approximation of Eq.1. The most common solution is the Monte Carlo approximation of the expectation as

$$\mathbf{X}_t^m = \mathbb{E}\left[\mathbf{x}_t|\mathbf{z}_{1:t}\right] \approx \sum_{j=1}^{N_s} w_t^j \mathbf{x}_t^i. \tag{4}$$

3.1. Filter Implementation

Two crucial factors are to be taken into account when implementing a PF: the likelihood evaluation and the propagation model. For a given target m, an adaptive reference histogram $\mathbf{H}_t^m$ of the colored surface voxels is available. CbCr color space is chosen due to its robustness against light variations and 21 bins for every channel are employed in the calculations. Let $\mathcal{E}_t^j$ be the 3D ellipsoid centered at $\mathbf{x}_t^j$ with a fixed size roughly modelling the human body. Function $p(\mathbf{z}_t|\mathbf{x}_t^j)$ can be defined as the likelihood of the ellipsoid $\mathcal{E}_t^j$ overlapping the volume corresponding to the tracked person and matching its color histogram. Information obtained at time t from the binary, $\mathbf{V}_t^b$, and color, $\mathbf{V}_t^c$, 3D reconstructions (see Figs. 1c and 1d) are used to define the likelihood function as:

$$p(\mathbf{z}_t|\,\mathbf{x}_t^j) = \alpha \frac{|\mathbf{V}_t^b \cap \mathcal{E}_t^j|}{|\mathcal{E}_t^j|} + (1-\alpha) B(\mathbf{H}_t^m, H(\mathbf{V}_t^c \cap \mathcal{E}_t^j)), \tag{5}$$

where $|\cdot|$ is the number of occupied voxels of the enclosed volume, $B(\cdot)$ is the Bhattacharya distance and $H(\cdot)$ stands for the color histogram extraction operation. Factor α controls the influence of each term (foreground and color information) in the overall likelihood function. Value $\alpha = 0.5$ provided satisfactory results.

Propagation model has been chosen to be a Gaussian noise added to the state of the particles after the re-sampling step. More sophisticated schemes employ previously learned motion priors to drive the particles more efficiently [1]. However, this would penalize the efficiency of the system when tracking unmodelled motions patterns and, since the proposed algorithm is intended for any type of motion, no dynamical model is adopted.

4. SPARSE SAMPLING

PF approach to tracking defines a set of instances of the position of the tracked person, the particles, and a formulation to measure the fitness of these hypothesis with relation to the observable data. However, the evaluation of this likelihood function may be computationally expensive. An alternative to PF is devised by reviewing the estimation of the state $\mathbf{X}_t$ in Eq.4. Centroid of the person may be alternatively extracted by computing the expectation over all the surface voxel positions. By randomly selecting a given number of voxels on this surface, it is still possible to obtain an enough accurate estimation of $\mathbf{X}_t$. We define the *sparse sampling* (SS) algorithm as a method to recursively estimate $\mathbf{X}_t$ from an evolving set of samples placed on the surface of the tracked person. Since we are no longer exploring the state space, we will talk about *samples* instead of *particles*.

Essentially, the proposed algorithm follows the PF analysis loop (re-sampling, propagation, evaluation and estimation). Being our volume a discrete representation, samples are constrained to occupy a single voxel and move with displacements on the 3D discrete orthogonal grid. By defining the appropriate likelihood function, samples attain high weights when placed on the surface while the re-sampling block is constrained to place the newly created samples on the foreground voxels. With this process, we define a recursive way to obtain a sparsely sampled version of the surface of the target and, therefore, its centroid.

4.1. Likelihood evaluation

Function $p(\mathbf{z}_t|\mathbf{x}_t)$ can be defined as the likelihood of a sample belonging to the surface corresponding to a target characterized by an adaptive reference color histogram $\mathbf{H}_t^m$. Let $\mathcal{C}(\mathbf{x}_t^j, q)$ be a neighborhood over a connectivity q domain on the 3D orthogonal grid around a sample placed in voxel $\mathbf{x}_t^j$. Then, we define the occupancy and color neighborhoods around $\mathbf{x}_t^j$ as $\mathbf{O}_t^j = \mathbf{V}_t^b \cap \mathcal{C}(\mathbf{x}_t^j, q)$ and $\mathbf{C}_t^j = \mathbf{V}_t^c \cap \mathcal{C}(\mathbf{x}_t^j, q)$, respectively. For a given sample j occupying a voxel, its likelihood may be formulated as

$$p(\mathbf{z}_t|\,\mathbf{x}_t^j) = \alpha\left(1 - \left|\frac{2|\mathbf{O}_t^j|}{|\mathcal{C}(\mathbf{x}_t^j,q)|} - 1\right|\right) + (1-\alpha)D(\mathbf{H}_t^m, \mathbf{C}_t^j), \tag{6}$$

where the first term measures the likelihood of a sample being placed in a surface voxel, attaining its maximum value when the half of its neighborhood is occupied. Since $\mathbf{C}_t^j$ contains only local color information with reference of the global histogram $\mathbf{H}_t^m$, the second term employs a color distance $D(\cdot)$ able to measure such color similarity. For every voxel in $\mathbf{C}_t^j$, it is decided whether it is similar to $\mathbf{H}_t^m$ by selecting the histogram value for the tested color and checking whether it is above a threshold γ 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 γ is computed so that the 80% of the values of $\mathbf{H}_t^m$ are taken into account. In our research $q = 26$ provided accurate results.

4.2. 3D Discrete Re-sampling

The re-sampling step has been defined according to the condition that every sample is assigned to a foreground voxel. In other words, re-sampling has usually been defined as a process where some noise is added to the position of the re-sampled particles according to their weights [7]. The higher the weight, the more replicas will be created. In our current tracking scenario, re-sampling adds some *discrete* noise to samples only allowing motion within the 3D discrete positions of adjacent foreground voxels as depicted in Fig.2a. Then, non populated foreground voxels are assigned to re-sampled samples. In some cases, there are not enough adjacent foreground voxels to be assigned, then a connectivity search finds closer non-empty voxels to be assigned as shown in Fig.2b.

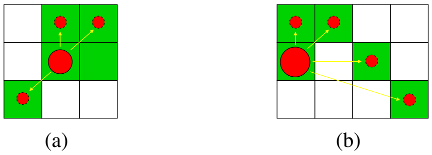

Fig. 2. Discrete re-sampling example (in 2D).

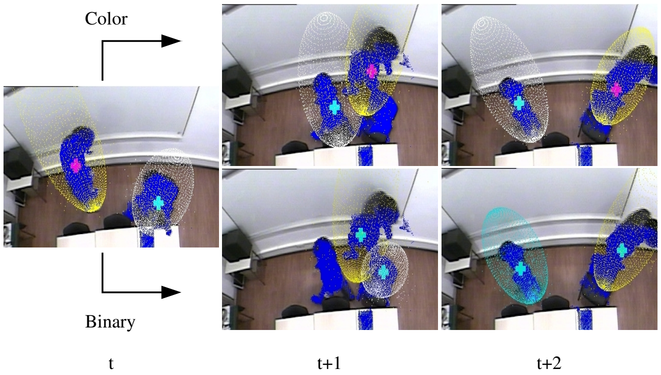

Fig. 3. Zenital view of two comparative experiments showing the influence of color in the SS algorithm when there is a cross-over between two targets (white and yellow ellipsoids).

5. MULTI-PERSON TRACKING

The proposed solution for multi-person tracking is to use a split tracker per person together with an interaction model. Let us assume that there are M independent trackers. Nevertheless, they are not fully independent since each tracker can consider voxels from other targets in both the likelihood evaluation or the 3D re-sampling step resulting in target merging or identity missmatches. In order to achieve the most independent set of trackers, we consider a blocking method to model interactions. Many blocking proposals can be found in 2D tracking related works [1] and we extend it to our 3D case. Blocking methods penalize particles/samples that overlap zones with other targets. Hence, blocking information can be also considered when computing the particle weights as:

$$w_t^j = p(z_t|x_t^j) \prod_{\substack{k=1\\k\neq m}}^{M} \beta\left(X_{t-1}^m, X_{t-1}^M\right), \tag{7}$$

where M is the total number of trackers, m the index of the evaluated tracker and X is the estimated state. Term $\beta(\cdot)$ is the blocking function defining exclusion zones that penalize particles that fall into them. For our particular case, considering that people in the room are always sitting or standing up (this is a meeting room so we assume that they never lay down), a way to define an exclusion region modeling the human body is by using an ellipsoid with fixed x and y axis. Axis in z is a function of the estimated centroid height. Tracked objects that come very close can be successfully tracked even though their volumes have partially merged.

Filtering spurious objects that appear in scene reconstruction and discarding non-relevant objects such as chairs or furniture is managed by the last module of the system that performs a higher semantic analysis of the scene.

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we collected a set of multi-view scenes in an indoor scenario involving up to 6 people, for a total of approximately 25 min. The analysis sequences were recorded with 5 fully calibrated and synchronized cameras with a resolution of 720x576 pixels at 25 fps (see a sample in Fig.1). The test environment is a 5m by 4m room with occluding elements such as tables and chairs. Groundtruth data was labelled manually allowing a quantitative measure of tracker's performance.

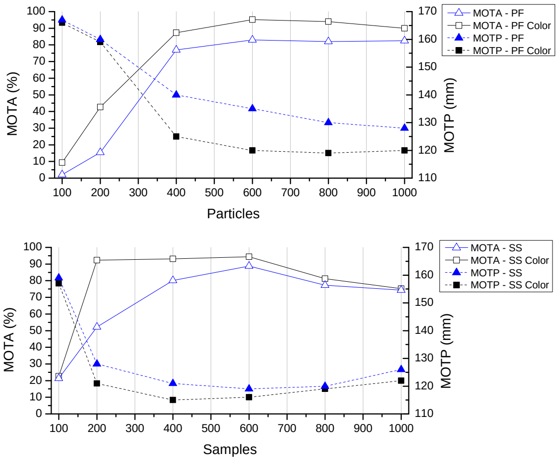

Fig. 4. $MOTP$ and $MOTA$ scores for various number of particles/samples comparing the Particle Filtering and the Sparse Sampling algorithms. Color influence is also depicted. Low $MOTP$ and high $MOTA$ scores are preferred indicating low metric error when estimating multiple target 3D positions and high tracking performance.

It should be noted that the employed test database has been included in the CLEAR06 Evaluation [9].

Metrics proposed in [11] for multi-person tracking evaluation have been adopted. These metrics, being used in international evaluation contests [9] and adopted by several research projects such as the European CHIL or the U.S. VACE allow objective and fair comparisons. Two metrics employed are: the **M**ultiple **O**bject **T**racking **P**recision (*MOTP*), which shows tracker's ability to estimate precise object positions, and the **M**ultiple **O**bject **T**racking **A**ccuracy (*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 two proposed systems where tested and the results reported in Table 1. PF and SS achieved similar performances but the main difference arose in the computational load measured as frames per second (fps). Since likelihood function is computed over a local neighborhood in the SS case, the overall complexity of the algorithm is reduced in comparison with the PF. Furthermore, the impact of color information in $MOTP$ and $MOTA$ scores for the two proposed tracking systems is depicted in Fig.4.

7. CONCLUSIONS AND FUTURE WORK

This paper presented two multi-person tracking systems in a multiple camera view environment. Redundant information among cameras is exploited to produce a 3D information that is employed by the proposed trackers. A PF based strategy proved efficient for this task but requiring a high computational load. In order to alleviate such drawback, sparse sampling technique has been presented as an alternative producing similar results but demanding roughly half of the processing time. Color information together with a blocking scheme has been employed to model interactions among targets thus adding robustness against mismatches and cross-overs among targets in both systems. Promising results obtained over a large test database proved the effectiveness of our techniques. Future research involves integration with audio technologies towards a multimodal tracking system.

	PF			PF Color		
Particles	MOTA	MOTP	fps	MOTA	MOTP	fps
100	2.1	167	0.95	9.3	166	0.96
200	15.4	160	0.75	42.7	159	0.69
400	77.2	142	0.38	87.3	125	0.36
600	**83.0**	**135**	**0.29**	**95.2**	**120**	**0.25**
800	81.9	131	0.27	94.0	119	0.20
1000	82.3	128	0.22	90.1	120	0.15

	SS			SS Color		
Samples	MOTA	MOTP	fps	MOTA	MOTP	fps
100	21.3	159	1.80	22.6	157	0.92
200	52.2	128	2.24	92.3	121	0.90
400	80.1	121	2.14	93.1	115	0.88
600	**88.8**	**119**	**1.47**	**94.4**	**116**	**0.72**
800	81.2	120	1.57	77.3	119	0.77
1000	74.3	126	1.42	75.2	122	0.70

Table 1. Quantitative experiments for different number of particles/samples. Voxel size was set to be $\nu = 2$ cm.

8. REFERENCES

[1] Z. Khan, T. Balch, and F. Dellaert, "Efficient particle filter-based tracking of multiple interacting targets using an MRF-based motion model," in *Int. Conf. on Intelligent Robots and Systems*, 2003, vol. 1, pp. 254–259.

[2] A. Pnevmatikakis and L. Polymenakos, "2D person tracking using Kalman filtering and adaptive background learning in a feedback loop," in *Lecture Notes on Computer Science*, 2007, vol. 4122, pp. 151–160.

[3] K. Bernardin, T. Gehrig, and R. Stiefelhagen, "Multi and single view multiperson tracking for SmartRoom environments," in *CLEAR Evaluation Workshop*, 2006.

[4] O. Lanz, "Approximate Bayesian multibody tracking," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 28, no. 9, pp. 1436–1449, 2006.

[5] C. Canton-Ferrer, J. R. Casas, and M. Pardàs, "Towards a Bayesian approach to robust finding correspondences in multiple view geometry environments," in *Lecture Notes on Computer Science*, 2005, vol. 3515, pp. 281–289.

[6] G.K.M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler, "A real time system for robust 3D voxel reconstruction of human motions," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 714–720.

[7] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Tran. on*, vol. 50, no. 2, pp. 174–188, 2002.

[8] A. Lopez, C. Canton-Ferrer, and J.R Casas, "Multi-person 3D tracking with particle filters on voxels," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2007.

[9] "CLEAR Evaluation," http://www.clear-evaluation.org.

[10] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 252–259.

[11] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *IEEE Int. Workshop on Vision Algorithms*, 2006, pp. 53–68.