Multicamera People Tracking Using a Locus-based Probabilistic Occupancy Map

Tao Hu[1,2], Sinan Mutlu[1,2], and Oswald Lanz[1]

[1] FBK Fondazione Bruno Kessler, Via Sommarive 18, I-38123 Povo(TN), Italy
[2] ICT Doctoral School, University of Trento, I-38123 Povo(TN), Italy
{hutao,mutlu,lanz}@fbk.eu

Abstract. We propose a novel people detection method using a Locus-based Probabilistic Occupancy Map (LPOM). Given the calibration data and the motion edges extracted from all views, the method is able to compute the probabilistic occupancy map for the targets in the scene. We integrate the algorithm into a Bayesian-based tracker and do experiments with challenging video sequences. Experimental results demonstrate the robustness and high-precision of the tracker when tracking multiple people in the presence of clutters and occlusions.

Keywords: people detection, multicamera tracking, probabilistic occupancy map.

1 Introduction

Currently people detection and tracking is a hotspot in the domain of computer vision due to its broad envisaged applications, which runs the gamut from Ambient Assisted Living, Security and Surveillance, Traffic Monitoring to Human Computer Interaction and Sports Analysis, etc (see [10] as a reference). Among all current tracking methods, visual tracking or camera-based tracking has many advantages over other methods such as laser-based tracking [11], since it is less intrusive and video sequences encode much richer information about the observed scene. Current visual tracking approaches fall into two categories: monocular approaches and multicamera approaches, according to the number of views they use. While monocular approaches have the advantage of simple and easy deployment but suffer from relying on limited 3D information and thus often fail in case of long period of full occlusions under unpredictable motions, multicamera approaches are used to tackle these problems by fusing complementary observations from multiple sensors. However, tracking multiple people robustly in highly crowded and cluttered scenes with affordable computational cost is still challenging.

Most of the state-of-the-art tracking systems adopts a probabilistic framework, as it takes into account uncertainties and ambiguities arising from either noise or occlusions and clutters of the scenes. Probabilistic tracking methods are based on the Bayesian filtering framework, which transform the tracking problem to a process of posterior probability density estimation of the states

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 693–702, 2013.

(e.g., people's locations, orientations and other motion parameters). To estimate the posterior probability density, a common method used is the particle filter (PF), which was initially used in visual tracking and dubbed as the Condensation algorithm in [5]. When PF is applied to visual tracking, there comes an inevitable issue of how to detect newly-entered objects in the scene. Traditional blob-based methods often have the trouble of isolating different objects in a merged blob and are prone to failures in the presence of high-rate occlusion. This has brought out some new methods like probabilistic occupancy map [3] or similar methods which projects features on multiple parallel planes [1,4].In this paper, we present a multicamera people tracking framework, the core of which is a locus-based probabilistic occupancy map. Given the motion edges in each camera view and the calibration data, we are able to compute the probabilistic occupancy map of the ground plane for the targets online.

The remainder of the paper is structured as follows. Section 2 introduces the state-of-the-art work to our interest. Section 3 presents the structure of our tracking system. Section 4 details how we compute the locus-based probabilistic map and how we implement tracking based on it, which is our main contribution. In Section 5 we test our algorithm with some data sets and give the results and analysis. And we draw conclusions in Section 6.

2 Related Work

Across literature, many detection and tracking methods have been proposed by exploiting features such as color, shape, contour or a combination of them. In [7], a Bayesian Multi-Blob Tracker based on statistical appearance model is proposed. Tracking is performed using one camera and thus has the problem of confusion when one object passes the other. In [8], a color distribution model using polar representation is proposed for multicamera tracking. Since color information is often corrupted by noise and easily get biased due to illumination changes and shadows, a combination of color information with other cues like edges or contours can be used such as the method in [2].

Recently occupancy map methods have gained popularity. Most commonly these methods back-project object models to a discretized occupancy map and compute the probability of objects being in the map. In [6], a people counting method is proposed by using the projection of visual hull intersections, which is generated by projecting the silhouette cones from each view. Fleuret et. al.[3] used a generative model of simple rectangles with the height of a typical pedestrian placed at given locations to simulate occupancy and back-projected the rectangles to the camera views. The probability of occupancy at every position is then approximated by minimizing the Kullback-Leibler distance between the marginals of a product law and the conditional posterior distribution. Although the algorithm can effectively estimate the occupancy on the ground plane for each frame independently given the output of a simple background subtraction algorithm, it suffers when there are too many people in the scene with serious occlusions for background subtraction algorithms to resolve individual blobs.

Khan and Shah [1] proposed a method based on geometrical constructs to resolve occlusion, the core of which is the planar homographic occupancy constraint. The constraint states that if a foreground pixel's piercing point (the intersection point of the ray casting from the pixel to the reference plane) falls inside a foreground object, then the same pixel will warp to the foreground region in all views. By exploring this constraint and gathering evidence from all the views into a synergistic framework, they can compute the likelihood of the scene location being occupied by foreground objects. The method can detect multiple people reliably in many complex scenarios but, according to the author, may cause false negative and false positive under some extreme situations due to the fact that it is purely image-based. A most recent related work can be found in [4], where a 3-D Marked Point Process model was proposed to detect and localize people.The method extracts pixel-level features by projecting foreground silhouettes on the ground plane and the hypothetical head plane, and estimates the positions and heights of the objects using a global optimization process. In [2], a detection method using motion edges is proposed. When a new object enters the scene, hypothetical samples are drawn from an informed detection prior, which is derived as the inverse of motion likelihood. A major drawback of this method is that it needs to compute a likelihood look-up table offline for all pixels in all images. Unlike [2], our method is able to generate online the possible states of the object for a given motion pixel. We compute the locus for all motion pixels in all views and integrate them to generate an occupancy map.

3 Problem Formulation

Our tracking system is based a Bayesian filtering framework. This section describes how we model the scene.

3.1 Bayesian Framework of Our Tracking System

The core idea of Bayesian filtering is to construct a posterior probability density function (PDF) of the system state with known information, i.e. given a set of observations z_t, to iteratively calculate the posterior density $p(x_t|z_t)$, where x_t denotes the system state at time t. The process consists of two steps: prediction and updating. First predict the state variables via the transition equation of the system, and then update the predicted values with the latest observation, which are represented by the following two equations respectively.

$$p(x_t|z_{1:t-1}) \propto \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})d_{x_{t-1}} \tag{1}$$

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|z_{1:t-1}) \tag{2}$$

Due to the complexity of the likelihood function $p(z_t|x_t)$ in practice, the posterior can hardly be expressed in closed form. Therefore, particle filtering was proposed to give an approximate solution to Bayesian filtering. The basic idea is to construct a posterior density based on a set of samples. The posterior $p(x_t|z_t)$ is represented by a set of particles with weights$\{x_t^i, w_t^i\}_{i=1}^N$. In this way, the posterior density at time t can be approximated by: $p(x_t|z_t) \approx \sum_{i=1}^N w_t^i \delta(x_t - x_t^i)$. To track a number of objects, a joint particle filter could be used. However, the number of particles grows exponentially with the number of objects, which in practice makes the computation intractable. One sub-optional solution is to use the Hybrid Joint-Separable filter [9], which has a quadratic complexity and is also used in this paper.

3.2 Likelihood Model and Motion Model

In our tracking framework, the state of the object incorporates the 2D position of the object on the ground plane, the heights and widths of the object, and a color model for the dominant parts (head, torso and legs). For a given time t, the state of the scene can be represented by a vector $x_t = \{k_t, x_t^1, x_t^2, ..., x_t^{k_t}\}$, where k_t is the number of targets at time t. The likelihood model specifies how to measure the match between the projected synthetic image for a given hypothetical state and the input image. To build a robust likelihood model, we take into account both the match of edges and color information. The motion model specifies how the current state is expected to propagate to the next time instance. Tracking multiple targets consistently in the scene entails us considering not only how the state of a target evolves from the current state to the next state $(p(x_t^i|x_{t-1}^i))$, how the targets interacts with each other $(p(x_t^{i_1}, x_t^{i_2}))$, but also how many new targets are likely to appear in or disappear from the scene $(p(k_t|k_{t-1}))$, and what are the likely configuration of a newly entered target j $(p(x_t^j))$. Therefore, the final joint scene state can be represented by a sparse dynamic graphical model and takes the form of [2]:

$$p(k_t|k_t - 1) \cdot \prod_{\xi_t \times \xi_t} p(x_t^{i_1}, x_t^{i_2}) \cdot \prod_{\xi_{t-1}} p(x_t^i|x_{t-1}^i) \cdot \prod_{\xi_t|\xi_{t-1}} p(x_t^j) \tag{3}$$

where ξ_t is the target index set at time t. Among the four items in (3), the dynamic model $p(x_t^i|x_{t-1}^i)$ and the interaction model $p(x_t^{i_1}, x_t^{i_2})$ can be designed a prior (e.g.,by a Gaussian model and proximity exclusion prior). However, the modeling of $p(k_t|k_t - 1)$ and $p(x_t^j)$ is rather difficult in an open scene where people can come and go randomly. In this paper, we attempt to address this problem as our main contribution. In general, we use a locus-based probabilistic occupancy map to detect newly entered targets, which will be detailed in the following section.

4 Detection Using Locus-based Occupancy Map

This section introduces how we generate the probabilistic occupancy map given motion images from all calibrated views. The general idea is, for a given motion pixel in each view, we compute all the possible states (or the locus) of the object in the scene that activate the motion pixel using geometric constraints. Then we accumulate the locus on the ground plane and compute the probabilistic occupancy map using kernel density estimation.

4.1 State Space Framework for Object Detection and Tracking with Parametric Shape Model

To reduce the computational burden while keeping a reasonable approximation of the target, we adopt a primitive shape model (Fig.1(a)) which represents the upper body part of a person and is comprised of connected cones as used in [2]. The shape model has two degrees of freedom: the height and the width. Suppose the surface of a rigid object (such as our model) is given in a parametric form through a 2-dimensional manifold $S(u, v|\mathbf{x})$ embedded in the 3D Euclidean space ($u, v \in [0 : 1]$ are curvilinear coordinates), and $\mathbf{x}$ is the object state, for a vertically symmetric object with profile $\rho(z)$ and height h moving on the ground, we have

$$S(u, v|x_0, y_0) = \begin{cases} \rho(h \cdot u) \cdot \cos(2\pi \cdot v) + x_0 \\ \rho(h \cdot u) \cdot \sin(2\pi \cdot v) + y_0 \\ h \cdot u \end{cases} \tag{4}$$

in which (x_0, y_0) is the 2D position of the object on the floor. In the case of a model built with connected cones like the one we are using, the profile $\rho(z)$ is piece-linear with connected segments. In order to compute the locus of the object for a given motion pixel, here we exploit some geometric constraints. Suppose the ray casting from the optical center to the image plane is parameterized by $R(t|P, V) = P + t \cdot V$ where P is a point on the line (e.g. the optical center of a camera) and V is a direction vector[1], we have the following geometric constraints:

Touch Constraints. The touch constraints are imposed by requiring orthogonality between V and the surface normal $S^{\perp}(u, v|\mathbf{x}) = (\nabla_u \times \nabla_v)S(u, v|\mathbf{x})$ at ray-surface intersections. Therefore, we seek the set of target states satisfying $\langle S^{\perp}(u, v|\mathbf{x}), V\rangle = 0$ and $S(u, v|\mathbf{x}) = P + t \cdot V$, or equivalently

$$\begin{aligned}\textbf{locus}(P, V) = \big\{\mathbf{x} \mid \langle S(u, v|\mathbf{x}) - P, S^{\perp}(u, v|\mathbf{x})\rangle = \langle S^{\perp}(u, v|\mathbf{x}), V\rangle = \\ \langle S(u, v|\mathbf{x}) - P, S^{\perp}(u, v|\mathbf{x}) \times V\rangle = 0\big\}\end{aligned} \tag{5}$$

ideally with an appropriate parametrization. Equation (5) gives a formularized expression of *locus*. For the surface model defined in (4), we have

$$S^{\perp}(u, v) \propto \begin{cases} \cos(2\pi \cdot v) \\ \sin(2\pi \cdot v) \\ -\rho'(h \cdot u) \end{cases}, \quad S^{\perp}(u, v) \times V \propto \begin{cases} V_z \cdot \sin(2\pi \cdot v) + V_y \cdot \rho'(h \cdot u) \\ -V_z \cdot \cos(2\pi \cdot v) - V_x \cdot \rho'(h \cdot u) \\ -V_x \cdot \sin(2\pi \cdot v) + V_y \cdot \cos(2\pi \cdot v) \end{cases}$$

[1] Notation: (P) capital letters denote 3D points, (P_x) subscript denotes the x component of P, (u) lower case letters are scalars, $(\mathbf{x})$ bold letters are points in the target state space, (m) roman letters denote pixels, $\langle :, : \rangle$ is the internal and $\times$ the external product, ∇ is the differential operator.

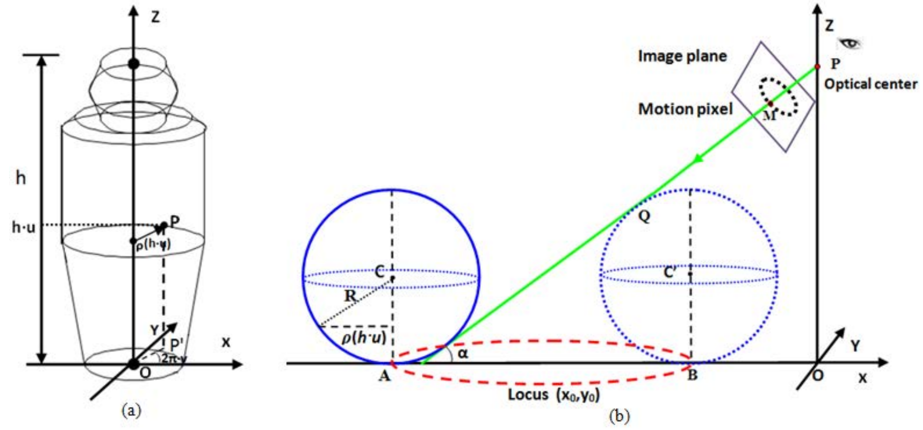


Fig. 1. (a)The object model we use in the paper is composed of a serial of connected cones. An arbitrary point P on the surface has the 3D coordinates $(\rho(h \cdot u) \cdot \cos(2\pi \cdot v), \rho(h \cdot u) \cdot \sin(2\pi \cdot v), h \cdot u)$ (b)The locus of a ball is an ellipse. XOY is the ground plane. PQ is the ray that goes from the optical center P and passes a motion pixel on the image plane. A and B are the farthest and nearest positions the ball can reach when it moves while keeping its surface tangent to the ray. The red dash curve is the trajectory (or locus) of the ball, which turns out to be an ellipse.

By exploiting the touch constraints in (5), we can derive the solution for locus , which results in a 1-dimensional manifold on the ground plane, as will be presented in the next section.

4.2 Parametric Form of Locus for a Vertically Symmetric Object on the Ground

Combining the surface equation in (4) and the constraints in (5),we get a linear-quadratic system of equations:

$$\begin{cases} \langle [V_x, V_y], [\mathrm{x}, \mathrm{y}] \rangle = \rho\rho' V_z \\ \langle [\mathrm{x}, \mathrm{y}], [\mathrm{x}, \mathrm{y}] \rangle = \rho^2 \end{cases} \tag{6}$$

in which (x, y) are the x and y coordinates of an arbitrary point on the line, satisfying $\mathrm{x} = P_x + t \cdot V_x - x_0$ and $\mathrm{y} = P_y + t \cdot V_y - y_0$. Under the assumption $V_x^2 + V_y^2 = \rho^2$ (this can be imposed through proper rescaling of V), the solution of locus is

$$\begin{cases} x_0(t) &= P_x - \langle [V_x, V_y], U(t) \rangle \\ y_0(t) &= P_y - \langle [V_y, -V_x], U(t) \rangle \end{cases} \tag{7}$$

where $U(t) = [\phi(t) - t, \pm\sqrt{1 - \phi^2(t)}]$, $\phi(t) = \frac{\rho' V_z}{\rho}$ with ρ, ρ' evaluated at $P_z + t \cdot V_z$. The following shows an example of the locus of a motion pixel with the profile of the object being a ball.

Algorithm 1. Locus-based Probabilistic Occupancy Map

```
locus set $\{x_i, w_i\}$; $w_i$ is the weight of the state $x_i$
for each camera $c$ do
  for each motion pixel $m$ do
    for height $h = h\_min; h < h\_max; h+ = step$ do
      {
      sample the height $h$ and compute the locus $\{x_c^m\}$;
      add $\{x_c^m, w_c\}$ to $\{x_i, w_i\}$; $//w_c$ is the weight of camera c
      }
    end for
  end for
end for
normalize weights $\{w_i\}$: $\sum_i w_i = 1$;
compute POM using Kernel for density estimation: $P_\sigma(x) = w_i \cdot \sum_i K_\sigma(x - x_i)$;
```

Example: Sphere Moving on the Ground. A most intuitionistic example is the locus of a sphere (Fig.1(b)). Suppose the optical center of the camera is at $P = (0, 0, H)$, the direction of the ray casting from P to the motion pixel m is $V = (1, 0, \tan\alpha)$ and the sphere is expressed by $\rho(z) = \sqrt{2Rz - z^2}$ (thus $\rho'(z) = (R - z)/\rho(z)$) with R being the radius , as is shown in Fig.1, the locus is an ellipse with equation: $(x_0 \sin\alpha + (H - R)\cos\alpha)^2 + y_0^2 = R^2$

For our shape model, the locus for a given motion pixel is a closed line of segments which form a stretched contour of the model. After computing the locus for all the motion pixels in all views (i.e., all the possible states of the object that can engender the motions in the views), we can generate the Probabilistic Occupancy Map (POM) for the object. A piece of pseudo code is shown in Algorithm.1.

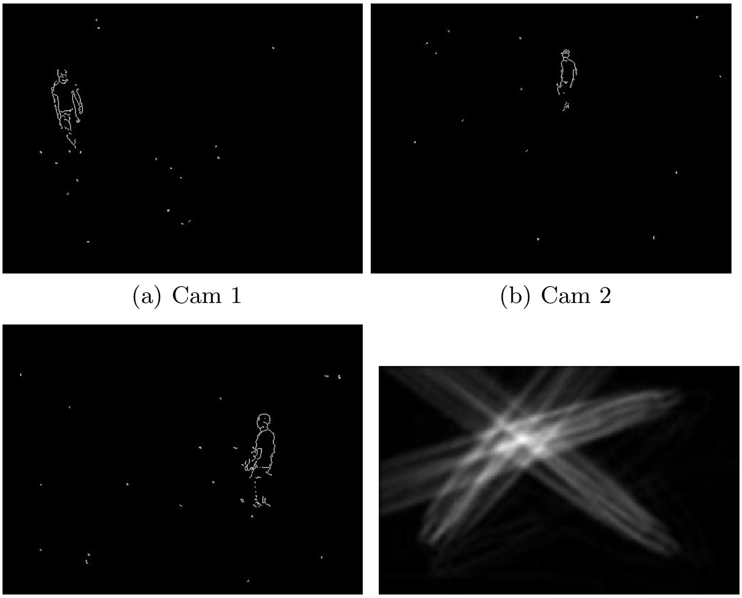
4.3 Computing Probabilistic Occupancy Map from Apparent Motion

The POM is computed by accumulating the locus of extracted motion pixels m $\in \mathcal{M}$

$$\mathrm{POM}(\mathbf{x}|\mathcal{M}) = \int_{\mathcal{M}} \mathrm{loc}(\mathbf{x}|\mathrm{m},)\, d\mathrm{m} \tag{8}$$

where $\mathrm{loc}(\mathbf{x}|\mathrm{m}) = \int K(\mathbf{x} - \mathbf{x}_0(t|P, V))\ dt$ is a kernelized probability density derived from Eq.(5) via convolution, with P, V obtained via back-projection of m. Eq.(8) assumes the height of the object is known. To account for different heights of the objects, we can compute the average POM by integrating over all possible heights ($h \in \mathcal{H}$) of a human:

$$\mathrm{POM}(\mathbf{x}|\mathcal{M}) = \int_{\mathcal{H}} \int_{\mathcal{M}} \mathrm{loc}(\mathbf{x}|\mathrm{m},)\, d\mathrm{m}\, dh \tag{9}$$



(a) Cam 1

(b) Cam 2

(c) Cam 3

(d) POM

Fig. 2. Motion edges from three views and the resulting POM

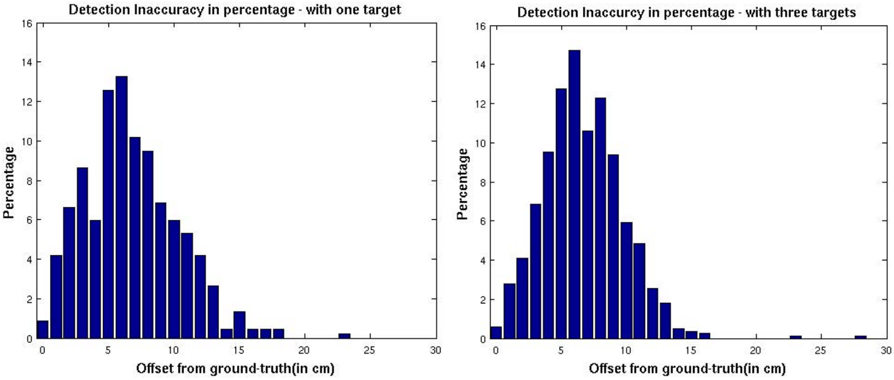


Fig. 3. A comparison of detection inaccuracy histogram between two sequences

4.4 Using Locus-based Probabilistic Occupancy Maps for Tracking

We have integrated the locus-based POM detection algorithm into our tracking system[9]. The tracker works as follows. First, motion edges are extracted from all the views using the Susan edge detector. With the motion edges, we generate the POM for the targets using the aforementioned algorithm. Then we search

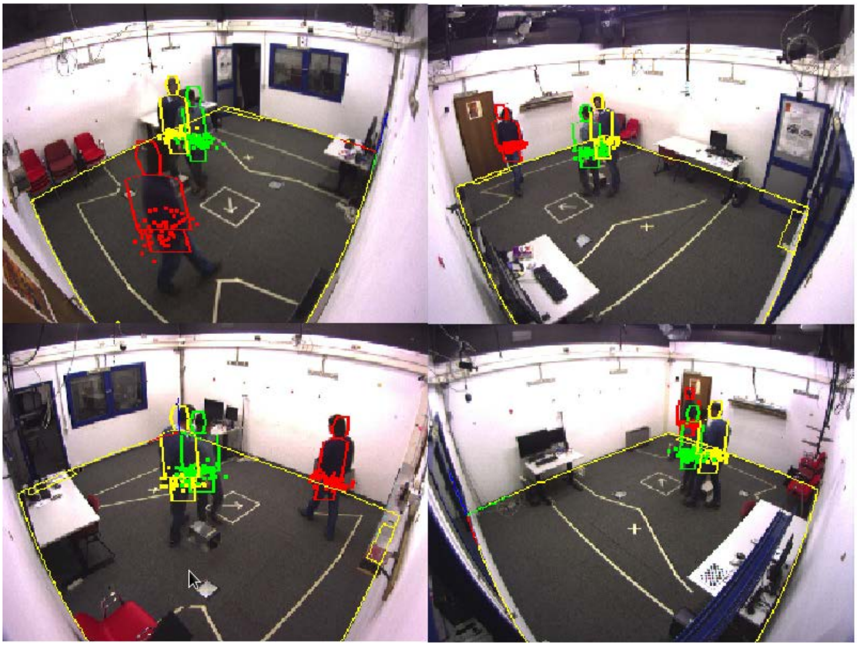


Fig. 4. Screenshots of the four views of our tracker. The dots around the shape models are particles (hypotheses).

the modes of the POM, new targets will be initialized where modes above a certain threshold.

5 Experiments and Results

We conducted experiments with two video sequences taken in our lab, which has a size of about 5×6 meters with four cameras mounted in the four corners of the room. The sequences were captured at 15 Hz with a resolution of 1024×768 pixels. The first sequence is about 3.5 minutes long with one person moving around the room. Fig.2 shows the motion edges detected in three of the four cameras and the resulting POM. As can be seen from Fig.2(d), the POM peaks at a point where the locus generated by the three views converges. We measured the detection inaccuracy histogram (Fig.3) by computing the distance between the detected position and the ground truth, which was done by manual labeling. We got a mean error of 0.072 meters. The other sequence is about 7 minutes with 3 persons moving around in the room, with constant occlusions. The resulting mean error is 0.076 meters, which does not increase much given the challenging scenes in the sequence. The results demonstrate that the detection algorithm is robust to clutters and occlusions. Fig.4 shows a screenshot of our tracker. The dataset used in the experiments and a short demo can be found at: http://tev.fbk.eu/DATABASES/MVPDT.html.

6 Conclusions

In this paper, we presented a novel people detection method using Locus-based Probabilistic Occupancy Map. We embedded the algorithm into a Bayesian-based tracker and did experiments with challenging video sequences. Experimental results demonstrate that the proposed approach can detect people robustly with high precision even in cluttered scenes with multiple people. If properly integrated with color based particle filtering, it leads to a robust solution for tracking multiple people in unconstrained scenarios. In future work we will explore further possibilities offered by the introduced method. In particular, we are interested in exploiting the typical shape of the locus-based POM for camera self-calibration using humans as a calibration pattern.

References

1. Khan, S.M., Shah, M.: Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. IEEE Transactions. PAMI 31(3), 505–519 (2009)
2. Lanz, O., Messelodi, S.: A Sampling Algorithm for Occlusion Robust Multi Target Detection. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, September 2-4, pp. 346–351 (2009)
3. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera People Tracking with a Probabilistic Occupancy Map. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 267–282 (2008)
4. Utasi, A., Benedek, C., Lengagne, R., Fua, P.: A Bayesian Approach on People Localization in Multicamera Systems. IEEE Transactions on Circuits and Systems for Video Technology 23(1), 105–115 (2013)
5. Isard, M., Blake, A.: CONDENSATION-conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
6. Yang, D.B., Gonzalez-Banos, H.H., Guibas, L.J.: Counting people in crowds with a real-time network of simple image sensors. In: Proceedings. Ninth IEEE International Conference on Computer Vision, October 13-16, vol. 1, pp. 122–129 (2003)
7. Isard, M., MacCormick, J.: BraMBLe: a Bayesian multiple-blob tracker. In: Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 2, pp. 34–41 (2001)
8. Kang, J., Cohen, I., Medioni, G.: Tracking people in crowded scenes across multiple cameras. In: Proc. 6th Asian Conf. Comput. Vision, pp. 390–395 (2004)
9. Lanz, O.: Approximate Bayesian multibody tracking. IEEE Trans. PAMI 28(9) (2006)
10. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surveys 38(4), 1–45 (2006)
11. Glas, D.F., Miyashita, T., Ishiguro, H., Hagita, N.: Laser-based tracking of human position and orientation using parametric shape modeling. Adv. Robot. 23, 405–428 (2009)