Fast Single-View People Tracking

Reza Hoseinnezhad[1], Ba-Ngu Vo[1], David Suter[2]
1: The University of Melbourne, Australia
2: The University of Adelaide, Australia

Abstract: This paper presents a fast people tracking technique comprising a simple background subtraction and Gaussian mixture PHD filtering on a constrained 3D motion model. Our technique is well-suited for video surveillance applications where behavior analysis and event detection are required without having to identify and track each individual. Comparisons with two state-of-the-art visual tracking methods and BraMBLe, a well-known recent technique, show that our method is substantially faster while exhibiting generally better tracking performance. The speed improvement is achieved by the use of simple background subtraction which saves computational resources and the exploitation of temporal information via PHD filtering, which compensates for the information loss incurred in the background subtraction at a fraction of the cost of a good detection scheme.

Keywords: visual tracking, background subtraction, object filtering, Bayesian estimation

1. Introduction

Visual tracking techniques commonly include three procedures focused on background/foreground modeling, detection, and filtering. In most techniques, *background/foreground modeling* results in a gray-scale image with the intensity of each pixel being its probability of belonging to background and the intensity values are thresholded to generate a binary image. *Detection* involves partitioning of the foreground pixels into connected regions in the image (also called blobs) and computing the location and size of each region. The *filtering* module estimates the number of moving objects and their states (properties such as location and size) from the results of detection.

Although visual tracking has been broadly studied in computer vision, most effort appears to have been focused on the "modeling" and "detection" modules. There is a large body of literature on models and techniques for background subtraction and object detection. The models are usually comprehensive enough to assure high accuracy in detection. Thus, with accurate detection and small rates of false alarms, almost any simple filtering method (or even no filtering) would result in sufficiently accurate tracking of the moving objects in the scene.

On the other hand, in military "target tracking" applications, detection is usually performed by simply thresholding the sensor (e.g. radar) measurements which usually results in numerous false measurements. There is a large body of literature on target tracking methods that are capable of estimating the number of targets and their states from measurements *immersed in clutter* (false positives or false alarms). The important task of "detection of the clutter and removing them from the state estimation process" is performed by a state-space filtering technique. This is in sharp contrast to many visual tracking methods where good detection using *sophisticated* background/foreground models results in low rates of false alarms. Filtering methods usually detect clutter based on the *temporal information* in the data. These information are mainly related to the dynamics of the target states and their evolution with time. To enable the filter to effectively detect clutter, we need *reliable* temporal information and therefore, *accurate* models for dynamics of targets states.

This paper shows that in common visual tracking applications, it is possible to make use of target dynamics to compensate for information loss in cheap background/foreground modeling. For people tracking, we propose a 3D motion model in which the people motion constraint on the floor is taken into account. Our motion model is more realistic than the motion models based in 2D blobs in image which are commonly used in the visual tracking literature.

Using our dynamic model and a nearly Bayes optimal filter (PHD filter), all the spurious object detections given by a simple background model can be removed and accurate estimates of locations and sizes of object blobs can be estimated. Most importantly, all the computations needed for background modeling, detection and filtering using a fast Gaussian mixture implementation of PHD filter – also called GM-PHD for short [1] – can run in a *real-time* frame rate by most modern computing platforms.

The output of the proposed visual tracking method includes all rectangular blobs containing people, their locations and sizes. Each blob may include more than one person in case occlusions occur, i.e. if the persons are too close to be distinguished in the detection results. The number of people in each blob is also estimated by the filter. We assume that for the desired event detection and behavior analysis, the system merely needs to estimate how many people are in the scene and where they are located. For such a task, our proposed tracking method suffices.

Our experiments show that as a result of the trade-off between expensive foreground/background modeling and filtering, our method significantly outperforms state-of-the-art methods in terms of computation. In addition, in terms

of estimation error, the performance of our tracking method is generally better than the examined methods.

2. Related work

Numerous sophisticated models and techniques have been developed for people tracking. In BraMBLe [2], background and foreground objects are modeled by Gaussian mixtures with their components trained offline. To enable the tracking to adapt to varying environmental conditions (e.g. ambient light) without frequent tuning, non-parametric models based on kernel density estimation techniques have been developed [3–5].

For the filtering module, many methods use sequential Monte Carlo (SMC) implementations of different multi-object filtering methods e.g. CONDENSATION in [2], an MCMC sampling approach for particle filtering in [6], a hierarchical particle filtering scheme in [7], Multiple Hypotheis Tracking (MHT) in [8] and a variational particle filter in [9]. To implement such techniques, we require to know what measurements are associated with each moving object. Due to its combinatorial nature, the data association problem requires heavy computational load.

The random finite set (RFS) approach to multi-target tracking is an emerging and promising alternative to the traditional association-based methods. In the RFS formulation, the collection of individual targets is treated as a *set-valued* state, and the collection of individual observations is treated as a set-valued observation. Modeling set-valued states and set-valued observations as RFS allows the problem of dynamically estimating multiple targets in the presence of clutter and association uncertainty to be cast in a Bayesian filtering framework. Novel RFS-based filters such as the Probability Hypothesis Density (PHD) filter [10] have generated substantial interest. The PHD filter has been recently used within a number of visual tracking solutions [11–16].

Similar to this work, in [11] the GM-PHD filter has been utilized for visual tracking where the filter is formulated for "color tracking" of objects with pre-specified color representations (pre-trained color histograms). Unlike this work, our technique is designed for visual surveillance applications where moving people in general (with previously unknown color patterns) are to be detected and tracked. Furthermore, the dynamic model used in [11] (i.i.d. random noise additives to state increments) is too simplistic for the tracking system to tolerate numerous false alarms. Indeed, the pre-trained color histograms help achieve close-to-perfect detections with minor false alarms. But, in visual surveillance applications, such information is not available and unless a sophisticated detection method is used, many false alarms will arise.

Our tracking technique is designed to work in real-time for typical visual surveillance applications where the video sequence images are taken by a *single* static video camera with known intrinsic parameters, height and angle of the camera with respect to the floor. Using the information, and the fact that people's feet are constrained to be on the floor (physical constraint on the location of their blobs in 3D world) and can be assumed straight standing most of the times (the physical height of each person does not vary with time substantially – another physical constraint), we develop a close-to-reality nonlinear model for target motion dynamics. This model is used as state transition model within the PHD filter.

3. Motion dynamic model

We will use lowercase letters to denote lengths and coordinates in the 2D image plane and uppercase letters for 3D world lengths and coordinates. Instead of the blobs in the image plane, we use blobs in the 3D world. But here, some important physical constraints are imposed to make the dynamic model more realistic than 2D image blob models commonly used in the visual tracking literature. First, each person is walking on the floor.[1] Therefore, each blob can be localized by only the Y and Z coordinates of its lower left corner in the floor plane, and its height and width (H, W). Variations of coordinates and size of a 3D world blob are modeled by the following constant-velocity model:

$$\begin{array}{rcl} Y(k+1) &=& Y(k) + T\dot{Y}(k) + \frac{T^2}{2} e_{_Y}(k) \\ \dot{Y}(k+1) &=& \dot{Y}(k) + T e_{_Y}(k) \\ Z(k+1) &=& Z(k) + T\dot{Z}(k) + \frac{T^2}{2} e_{_Z}(k) \\ \dot{Z}(k+1) &=& \dot{Z}(k) + T e_{_Z}(k) \\ H(k+1) &=& H(k) + T e_{_H}(k) \\ W(k+1) &=& W(k) + T e_{_W}(k). \end{array} \quad (1)$$

The parameter T denotes the sampling time and $e_{_{Y(k)}}$, $e_{_{Z(k)}}$, $e_H(k)$ and $e_W(k)$ are the noise samples representing small random accelerations for the location (Y, Z) on the floor, and random variations of the size of the blob (H, W), respectively.

Another constraint is based on the fact that people usually walk in an upright position, and their height does not vary as much as their location or width.[2] This constraint is imposed by setting the random variations of the height of the blobs ($\sigma_{_H}$) considerably smaller than the width ($\sigma_{_W}$) and location variations ($\sigma_{_Y}, \sigma_{_Z}$).

A third constraint is related to the designated locations of entrance and exit gateways for people. The information related to entrance and exit locations are effectively used by the GM-PHD filter as described in section 4.

3.1 Detection and measurement model

As mentioned previously in Section 1, background modeling results in a binary image which is processed in the detection step, and a number of blobs are returned as measurements. We are interested in measurements in the

[1] We focus on applications where the floor is a flat surface.

[2] The width of the blobs, W, can vary when people turn.

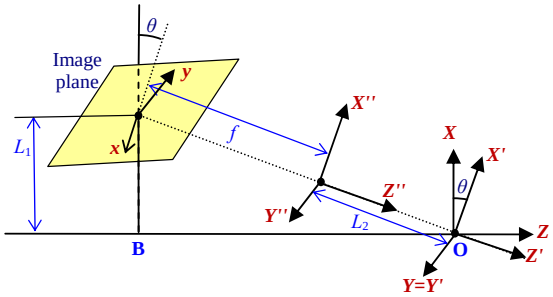


Figure 1: The relationship between a 3D world blob and a 2D blob in the image.

form of 3D world blobs and therefore, need to convert the location and size of a 2D image blob, denoted by $[x\ y\ h\ w]^\top$, to the location and size of its corresponding 3D world blob, denoted by $[Y\ Z\ H\ W]^\top$. Figure 1 shows how a blob $[Y\ Z\ H\ W]^\top$ undergoes a θ rotation to transform to $[Y^{'}\ Z^{'}\ H^{'}\ W^{'}]^\top$, then a translation to $[Y^{''}\ Z^{''}\ H^{''}\ W^{''}]^\top$, and a perspective transformation to the 2D blob $[x\ y\ h\ w]^\top$ in the image plane.

Let us denote the focal lengths of the camera (in pixels) by s_x and s_y, and the principal point coordinates of the camera by x_0 and y_0. These parameters can be easily estimated using a camera calibration method such as Bouguet's calibration toolbox [17] which we also used in our experiments.

Conversion of a 3D blob to a 2D blob in the image is then governed by the following equations:

$$x = s_x \frac{\sin(\theta)Z + \cos(\theta)X}{\cos(\theta)Z - \sin(\theta)X + L_2} + x_0 \quad (2)$$

$$y = s_y \frac{Y}{\cos(\theta)Z - \sin(\theta)X + L_2} + y_0 \quad (3)$$

$$h = s_x \frac{H}{\cos(\theta)Z - \sin(\theta)X + L_2} \quad (4)$$

$$w = s_y \frac{W}{\cos(\theta)Z - \sin(\theta)X + L_2}. \quad (5)$$

The "walking on the floor" constraint is imposed by $X = 0$,

$$\begin{array}{l} x = s_x \frac{\sin(\theta)Z}{\cos(\theta)Z+L_2} + x_0 \ ; \ h = s_x \frac{H}{\cos(\theta)Z+L_2} \\ y = s_y \frac{Y}{\cos(\theta)Z+L_2} + y_0 \ ; \ w = s_y \frac{W}{\cos(\theta)Z+L_2}. \end{array} \quad (6)$$

This equation expresses a one-to-one nonlinear relationship. A measurement $[x\ y\ h\ w]^\top$ in the image plane can be converted to a measurement $\mathbf{Z} = [Y\ Z\ H\ W]^\top$ in the 3D world,

$$\begin{array}{l} Z = \frac{L_2(x-x_0)}{s_x \sin(\theta) - \cos(\theta)(x-x_0)} \ ; \ H = \frac{(\cos(\theta)Z+L_2)h}{s_x} \\ Y = \frac{(\cos(\theta)Z+L_2)(y-y_0)}{s_y} \ ; \ W = \frac{(\cos(\theta)Z+L_2)w}{s_y}. \end{array} \quad (7)$$

To measure the extrinsic parameters L_2 and θ in equation (7), the point O on the floor (it corresponds to the principal point (x_0, y_0) in the image plane) is found and the distance OB and the height of the camera L_1 are measured. The extrinsic parameters are then given by:

$$\theta = \arctan\left(\frac{L_1}{OB}\right) \ ; \ L_2 = L_1 \csc(\theta) - f \quad (8)$$

where f is the focal length of the lens.

It is worth noting that constrained motion models for people tracking have previously appeared in the literature [18–20]. However, those models are designed for "multi-view" tracking and need the multi-view information for their calibration. Our model, in contrast, is based on single-view tracking and its parameters can be calibrated via on-spot measurements and camera calibration routines.

4. PHD filter

The PHD filter is a (near-optimal) multi-target Bayes filter in which the posterior intensity, the first moment of the posterior multi-target state, is propagated in time. Suppose that the number of blobs in 3D world (people) and their states are encapsulated in a random finite set X on $\mathcal{X}$. Its PHD is a non-negative function v on $\mathcal{X}$ such that for every $S \subseteq \mathcal{X}$, the expected number of elements of X in S is given by $\int_S v(x)\,dx$.

In the proposed Gaussian mixture implementation of the PHD filter, Vo and Ma [1] assume that the posterior PHD, v_{k-1} , at time $k-1$ is a Gaussian mixture of the form:

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \quad (9)$$

where J_{k-1} is the total number of Gaussian components and $w_{k-1}^{(i)}$, $m_{k-1}^{(i)}$ and $P_{k-1}^{(i)}$ are the weight, mean vector and covariance matrix of the i-th component. They derive a recursion to propagate the Gaussian mixture posterior PHD forward in time. We revised the recursion for our people tracking application as follows.

In addition to the matrix parameters of state transition and measurement models, F, Q, C and R, we need to model object births and deaths as well as detection uncertainty and false measurements as follows:

– a location-dependent survival probability:

$$p_S(x) = w_S^{(0)} + \sum_{i=1}^{J_S} w_S^{(i)} \mathcal{N}(x; m_S^{(i)}, P_S^{(i)}) \quad (10)$$

– a Gaussian mixture model for the PHD of the target birth process:

$$\gamma(x) = \sum_{i=1}^{J_\gamma} w_\gamma^{(i)} \mathcal{N}(x; m_\gamma^{(i)}, P_\gamma^{(i)}). \quad (11)$$

– a constant detection probability p_D;

– The PHD for clutters as a time-variant function $\kappa_k(z)$ of measurement z. Commonly this PHD is assumed constant, in which case it is the average number of false alarms per unit volume of the measurement space.

In equation (10), the weights, mean and covariance parameters ($w_S^{(i)}, m_S^{(i)}$ and $P_S^{(i)}$) are set in such a way that the survival probability is small at exit gateways and large elsewhere. In equation (11), the means $m_\gamma^{(i)}$ are the peaks of the birth PHD and have the highest local concentrations of expected number of spontaneous births, and represent the entrance gateways where people are most likely to appear. The covariance matrices $P_\gamma^{(i)}$ determine the spread of the birth PHD in the vicinity of the peak $m_\gamma^{(i)}$. The weight $w_\gamma^{(i)}$ gives the expected number of new people originating from $m_\gamma^{(i)}$.

Given the above parameters, the GM-PHD recursion is then implemented. The details of the recursion and a complete pseudo code can be found in [1]. Note that the number of Gaussian components will exponentially grow with time. To limit this number, a simple pruning procedure follows each recursion involving truncating and merging components.

The multi-target state estimates from the Gaussian mixture representation of the PHD are given by the local maxima of v_k which are the means of the constituent Gaussian components (if they are reasonably well-separated). Each mean will correspond to the state $[Y\ \dot{Y}\ Z\ \dot{Z}\ H\ W]^\top$ of a blob indicating a person or group of people. The total number of people in the scene is estimated by summing all weights of Gaussian components. The number of persons in the blob is given by rounding the weight of the Gaussian component to the nearest integer.

It is important to note that the number of Gaussian components in the GM-PHD filtering scheme is usually far less than the number of particles required in particle implementation of PHD filter and other techniques. The GM-PHD filter runs substantially faster than other filters in multi-target tracking applications. The number of Gaussian components in GM-PHD filter is commonly far less than the number of pixels or other small grid regions in each frame. The importance of this is revealed when we note that in Gaussian mixture or kernel density modeling techniques, there are several Gaussian components or density kernels for each pixel.

5. Experimental results

In a number of controlled visual tracking experiments, we have compared the performance of our technique with three recent methods that are based on detection using the following background models:

KDE: nonparametric modeling based on kernel density estimation [3] using the recent 20 frames;

GMM: four-component Gaussian mixture model [2] with their parameters tuned offline using the first 20 frames which include no moving targets;

SACON: short for sample consensus, this is a deterministic background model [21]. In our experiments, SACON also used the past recent 20 frames.

The models are implemented based on chromaticity colors (r, g) and intensity I of each pixel. In each experiment, the camera is calibrated to compute the intrinsic parameters, then the physical lengths L_1 and OB are manually measured to be used for computing the extrinsic parameters θ and L_2 using equation (8). In three experiments, two to four people (with different heights and appearances) moved in the same environment and the video sequence was recorded by the same camera (with the intrinsic parameters and the physical lengths L_1 and OB unchanged). For brevity, only the results of one experiment are presented in which four people enter the scene in different times, occlude and interact, then leave the scene.[3]

In our tracking method, the following simple background modeling is implemented. Using the first 20 frames, each pixel (i, j) is associated with three means and standard deviations denoted by $(\mu_{ij}^r, \mu_{ij}^g, \mu_{ij}^I)$ and $(\sigma_{ij}^r, \sigma_{ij}^g, \sigma_{ij}^I)$. Thus, only two rgI-colored images are recorded for the model. During the next frames in the image sequence, the pixel (i, j) is considered belonging to the background if

$$\exists\, c \in \{r, g, I\}\ :\ |c - \mu_{ij}^c| < \vartheta \sigma_{ij}^c \qquad (12)$$

where ϑ is a constant threshold to detect significant deviations in color or intensity. For the purpose of our experiments, we chose $\vartheta = 10$.[4] The background model can be adaptive to light variations and new static background objects by updating the two rgI-colored images as soon as there are no moving targets to track in the scene for a specified period of time (provided that the scene remains empty of targets for an extra 20 frames for the update).

Figure 2(a) shows the frame 167 out of 230 frames of this sequence of recorded frames. The binary image returned by our background model and the detected blobs are shown in Figures 2(b) and 2(c), respectively. The location and size of detected blobs are converted to 3D world blobs using equation (6), and the results given as inputs to the GM-PHD filter which in turn results in the blobs shown in Figure 2(d).

As it is observed in Figure 2(d), the number of people in each blob is also estimated by the PHD filter as it inherently keeps track of the number of people who enter and

[3] Note that evaluating our method on a publicly available database was not possible because the intrinsic and extrinsic parameters of the camera used in those databases are not known.

[4] For a single-mode distribution ϑ = 2.5 – 3.0 would be more appropriate, but to cater for the possible multimodal distributions we set a larger threshold.

Table 1: Comparative results for error measures and processing times of our methos and other examined techniques.

Method	e_{FA}	e_{FN}	Ave. Proc. Time (frames/sec.)
KDE	4.93%	0.00%	1.57
GMM	11.90%	2.05%	3.79
SACON	8.00%	2.98%	8.54
Our method	0.82%	0.51%	27.83

exit the scene (through the birth and death of targets as parametrized in Section 4). Without any filtering module in the other examined methods and by merely relying on detecting the objects by background models, each blob is naturally assumed to include only one person.

For a quantitative comparison, we have computed two error measures: a normalized false alarm denoted by e_{FA} and a normalized false negative (missed detections) denoted by e_{FN}, defined as follows:

$$e_{\text{FA}} = \frac{\text{\# of wrongly detected persons}}{\text{total number of frames}} \times 100\%$$
$$e_{\text{FN}} = \frac{\text{\# of missed persons}}{\text{total number of frames}} \times 100\%. \quad (13)$$

In addition, for all methods we have recorded the (averaged) processing time in terms of processed frames per second. We implemented all methods in MATLAB language on a Laptop with Intel(R) Core(TM)2 Duo CPU 2.53 GHz and 3.48 GB memory. The error measures and processing times are listed in Table 1. These results have been computed over a total of 974 image frames in a number of tracking experiments involving 1-4 people entering the scene and interacting with each other.

The results of Table 1 show that our method runs substantially faster than the other examined techniques while its error measures are generally better than them. The main reason for the increased speed is the saving of computation time via using a simple background subtraction. Let us denote the CPU processing times for a linear or comparison operation, and for an exponential (e.g. Gaussian pdf) computation, by τ_1 and τ_2, respectively. The image size of each frame is assumed $n_x \times n_y$ pixels. The KDE model processes each pixel of the current and $N_0 = 20$ recent frames for each color channel in Gaussian kernel density estimations and the major part of the computation time required by KDE model is given by: $T_{\text{KDE}} \approx 3N_0 n_x n_y \tau_2$. Similarly, for SACON model we have $T_{\text{SACON}} \approx 3N_0\ n_x n_y \tau_1$ and for the GMM, there are four Gaussian components and $T_{\text{GMM}} \approx 12\ n_x n_y \tau_2$. Our simple background modeling only needs computations of equation (12), $T_{\text{Simple Model}} \approx 3\ n_x n_y \tau_1$. Thus, the speed improvement is evidenced by noting that:

$$T_{\text{Simple Model}} \approx \frac{1}{N_0} T_{\text{SACON}} \approx \frac{1}{4}\frac{\tau_1}{\tau_2} T_{\text{GMM}} \approx \frac{1}{N_0}\frac{\tau_1}{\tau_2} T_{\text{KDE}}; \quad (14)$$

and the fact that $\tau_1 \ll \tau_2$. The above analysis is approximate and intended for indication purposes e.g. it does not take into account the relatively small amount of computation required by PHD recursions.

6. Conclusions

In this paper, we propose a fast people tracking method based on a simple background modeling followed by PHD filtering. The camera is assumed static and its intrinsic parameters and its height and angle with respect to the floor are measurable. We show that our method runs substantially faster than BraMBLe and two state-of-the-art visual tracking algorithms while exhibiting better tracking performance. Usage of a simple background subtraction saves computational resources, and temporal information is exploited by the PHD filter to compensate for the information loss incurred due to background subtraction. A 3D target dynamics model, that takes a number of physical constraints into account, allows the filter to make the best use of the temporal information in the data.

The proposed method has some limitations and restrictions. For example, in our target dynamics, we assume that the poses of people are upright and our GM-PHD filter works best if the entrance and exit gateways cover a relatively small area of the scene visible to the camera. We will work on resolving these issues in our future work.

Acknowledgements

This research was supported by Australian Research Council through the Discovery grant number DP0880553.

References

[1] B.-N. Vo and W.-K. Ma: *The gaussian mixture probability hypothesis density filter*, IEEE Trans. Signal Proc., 54(11):4091 – 104, 2006.

[2] M. Isard and J. MacCormick: *BraMBLe: a Bayesian multiple-blob tracker*, ICCV'01, volume 2, pages 34 – 41, Vancouver, British Columbia, Canada, 2001.

[3] A Elgammal, R Duraiswami, D Harwood, and Larry S. Davis: *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance*, Proceedings of the IEEE, 90(7):1151 – 1162, 2002.

[4] A. Tyagi, M. Keck, J. W. Davis, and G. Potamianos: *Kernel-based 3D tracking*, In CVPR'07, Minneapolis, Minnesota, USA, 2007.

[5] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis: *Sequential kernel density approximation and its application to real-time visual tracking*, PAMI, 30(7):1186–1197, 2008.

[6] K. Smith, D. Gatica-Perez, and J.-M. Odobez: *Using particles to track varying numbers of interacting people*, In CVPR'05, volume I, pages 962 – 969, San Diego, CA, USA, 2005.

[7] C. Yang, R. Duraiswami, and L. Davis: *Fast multiple object tracking via a hierarchical particle filter*, In ICCV'05, volume 1, pages 212 – 19, Beijing, China, 2005.

[8] S.-W. Joo and R. Chellappa: *A multiple-hypothesis approach for multiobject visual tracking*, IEEE Trans. Image Process., 16(11):2849 – 2854, 2007.

[9] Y. Jin and F. Mokhtarian: *Variational particle filter for multi-object tracking*, In ICCV'07, Rio de Janeiro, Brazil, 2007.

[10] B.-N. Vo, S. Singh, and A. Doucet: *Sequential monte carlo methods for multi-target filtering with random finite sets*, IEEE Trans. Aerospace and Elec. Sys., 41(4):1224 – 1245, 2005.

[11] Y.-D. Wang, J.-K. Wu, W. Huang, and A. A. Kassim: *Gaussian mixture probability hypothesis density for visual people tracking*, In Int. Conf. Information Fusion, Quebec City, Canada, 2007.

[12] Y. Wang, J. Wu, A. A. Kassim, and W. Huang: *Data-driven probability hypothesis density filter for visual tracking*, IEEE Trans. for Circuits and Systems for Video Tech., 18(8):1085–1095, 2008.

[13] E. Maggio, M. Taj, and A. Cavallaro: *Efficient multitarget visual tracking using random finite sets*, IEEE Trans. for Circuits and Systems for Video Tech., 18(8):1016 – 1027, 2008.

[14] Y.-D. Wang, J.-K. Wu, A. A. Kassim, and W.-M. Huang: *Tracking a variable number of human groups in video using probability hypothesis density*, In ICPR'06, volume 3, pages 1127 – 1130, Hong Kong, China, 2006.

[15] N. T. Pham, H. Weimin, and S.H. Ong: *Tracking multiple objects using probability hypothesis density filter and color measurements*, In IEEE Int. Conf. Multimedia and Expo, ICME'07, pages 1511 – 1514, Beijing, China, 2007.

[16] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro: *Particle phd filtering for multi-target visual tracking*, In ICASSP, volume I, pages 1101–1104, Honolulu, Hawaii, USA, 2007.

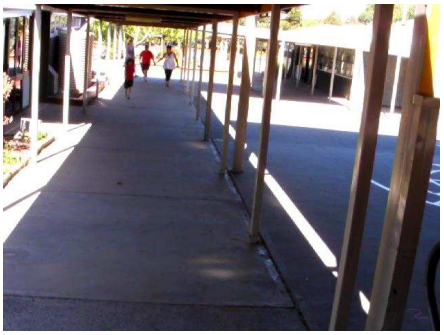
[17] J.-Y. Bouguet: *Camera calibration toolbox for matlab*, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html. Last updated June 2008.

[18] S. M. Khan and M. Shah: *A multiview approach to tracking people in crowded scenes using a planar homography constraint*, ECCV 2006, Part IV, LNCS 3954, pages 133–146. Springer-Verlag, 2006.

[19] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua: *Multicamera people tracking with a probabilistic occupancy map*, PAMI, 30(2):267–282, 2008.

[20] K. Kim and L. S. Davis: *Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering*, ECCV 2006, Part IV, LNCS 3954, pages 98–109. Springer-Verlag, 2006.

[21] H. Wang and D. Suter: *A consensus-based method for tracking: Modelling background scenario and foreground appearance*, Pattern Recognition, 40(3):1091 – 105, 2007.



(a)

(b)

(c)

(d)

Figure 2: (a) Frame 167 out of 230 frames of an experimental sequence (b) The binary image resulted by the background model (12) (c) The results of detections which include numerous spurious detections (d) The filter output, as the final tracking result.