Single and Multiple View Detection, Tracking and Video Analysis in Crowded Environments

Teng Xu [a], Peixi Peng [a], Xiaoyu Fang [a], Chi Su [a], Yaowei Wang [a,c+], Yonghong Tian [a+], Wei Zeng [b], Tiejun Huang [a]

[a] National Engineering Laboratory for Video Technology, School of EE & CS, Peking University

[b] NEC Laboratories, China

[c] Beijing Institute of Technology

[+]E-mail: ywwang@jdl.ac.cn yhtian@pku.edu.cn

Abstract

In this paper, we present our detection, tracking and event recognition methods and the results for PETS 2012. First, ROIs (Regions of Interest) based on geometric constraints are utilized in single view detection to eliminate the negative influence of clutter environment. Then, an optimized observation model is applied to address the ID switching or tracking drifting problem in single view tracking. Third, we introduce the multi-view Bayesian network (MBN) to reduce the "phantom" phenomena which frequently happen in general multi-view detection tasks. At last, a motion-based event recognition method is proposed to handle the event recognition task. Experimental results on the PETS 2012 dataset indicate that our methods are very promising.

1. Introduction

The challenge of PETS 2012 includes three different parts: 1) Estimation the count and density of crowd person; 2) Tracking individual(s) within a crowd; 3) Flow analysis and crowd events recognition. In this paper, we focus on the task 2) and 3).

Traditional detection methods suffer the performance degradation caused by clutter background and are often very time-consuming. Geometric constraint is a widely used kind of contextual information which could be utilized to generate ROIs [1-2]. With the ROIs, the search area of the detector could be limited to regions where pedestrians may appear. In this way, many background noises are dismissed. In this paper, we employ the ground plane and pedestrian height constraints to generate ROIs. Experimental results show that the detection accuracy and speed are improved obviously.

The ID switching or tracking drifting is the primary challenge of single view tracking task. In recent years, the tracking-by-detection [4-6] and multi-instance learning (MIL) [7] are two of the most widely used methods and could solve the ID switching problem to some extent but they also have their own drawbacks. The tracking-by-detection method cannot deal with the object occlusion problem, especially in the crowded scenes. And in the case of object occlusion, the outputs of object detection are often unreliable, consequently leading to the unreliable performance of tracking in these methods. As for the MIL, it can solve problems such as renewed faulty samples due to occlusion of a single object. However, when we track various objects and build a classifier for each of them, the classifier of a tracker may join the positive samples of another object, leading to ID switching or tracking drift. To address these problems, we propose a robust tracking-by-detection approach with an optimized observation model. Our observation model of particle filter fuses the detection results and three states of trackers in a unified probabilistic framework, where each state is represented with a MIL classification model. These states include the Original State that denotes the initial state of a tracker in the video, the Current State that characterizes the tracker's state at the present time, and the Max-Difference State that represents the state of the tracker which can best capture the appearance of the object and thus is most different from other trackers. Therefore, three state classifiers can be utilized to recognize the current status of each tracker. In our experiments, ID switching and tracking drift can be effectively avoided.

General multi-view detection methods will cause many "phantom" phenomena. Phantoms are the intersections of viewing rays at locations which are not occupied by any pedestrians [8]. Phantoms and pedestrians are occluded by each other. The occlusion relationship takes different influence to the pedestrians and phantoms. Hence the key problem to reduce the possible phantoms in the multi-view projection is how to effectively model and utilize the occlusion relationship among potential pedestrians at different locations in all views. To solve this problem, we introduce the multi-view Bayesian network (MBN) to model the potential occlusion relationship of all locations in all views. Moreover, we also model the "subjective supposing" node states (SSNS) as a set of Boolean parameters of MBN, which are then used to denote whether a pedestrian occurs at the locations. A learning algorithm is then proposed to estimate the SSNS parameters by finding the configuration that make the final occupancy possibility best explain the image observations (foreground masks) from different views. In addition, we use the single view

978-0-7695-4797-8/12 $26.00 © 2012 IEEE
DOI 10.1109/AVSS.2012.91

tracking information in our learning method as constraints such as the number of pedestrians.

Recognizing crowd events in video sequences involves both motion magnitude (e.g. walking vs. running) and orientation (e.g. splitting) information. Meanwhile, relationships between moving pedestrians are useful as well (e.g. merging and dispersion). In this paper, histograms of local motion velocities are computed for each frame in both rectangular coordinate and polar coordinate system. That means absolute as well as relative motion orientation and magnitude are encoded into the histogram feature. Using a SVM approach, crowd events are recognized easily by classifying these histogram features frame by frame.

The rest of this paper is organized as follows. Section 2, 3, 4 present our single view and multi-view detection, tracking and event recognition method in details. In Section 5, experimental results on PETS 2012 are reported. Finally in Section 6, conclusions are made.

2. Single View Detection and Tracking

2.1. Detection with geometric constraints

Clutter background may cause the performance degradation of detection methods. To solve these problems, we employ geometric constraints to generate regions of interest (ROIs). Firstly, the ground plane is divided into *N* levels in depth direction. Let l_{p1p2} be a line on the maximum-depth boundary of level l_p .The camera coordinate of p_1 is $(x_{cam}, y_{cam}, z_{cam})$ and the world coordinate is $(x_{world}, y_{world}, z_{world})$. When the ground is sloping, the world coordinate should satisfy the constraint: $\mathrm{L}(x_{world}, y_{world}, z_{world}) = 0$ and when it is horizontal, y_{world} is 0. The situation that the ground plane is not flat is outside the scope of this article. y_{cam} could be computed according to (1):

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = R\left(\begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} - C\right) \tag{1}$$

where *R* is the rotation matrix of the camera and *C* is the camera centre. Then we map p_1 to image coordinate system as (2).

$$\lambda \begin{bmatrix} x_{img} \\ y_{img} \\ 1 \end{bmatrix} = K \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \tag{2}$$

K is the internal parameter of the camera and λ is the normalized parameter. For simple calculation, we assume that the feet of pedestrian are on the ground plane and the height of general pedestrian should be in range of [1.1-2.0] meters. Let l_{p3p4} be a line on the other boundary of level l_p . For the ROI whose bottom is on the line l_{p1p2} , we could get the boundary of its top in camera coordinate system by plugging the maximum and minimum of pedestrian heights into (1) as the y_{world}. Combined with (2), two boundary lines in the image l_{1max} and l_{1min} could be obtained. For the ROI whose bottom is on the line l_{p3p4} , we can also get l_{2max} and l_{2min} in the same way. The boundary of the ROI's bottom l_{b1} and l_{b2} could also be obtained by setting y_{world} to 0. When the number of depth levels *N* is large enough, l_{1max} and l_{2max} would be very near. The same thing is true for l_{2min} with l_{1min} and l_{b1} with l_{b2} . For simple calculation, we set the bottom boundary l_b as l_{b2} , the top boundary l_h as l_{1max} and the bottom boundary l_l as l_{2min} . With these three lines, ROIs could be generated. Thus many background noises are dismissed and the detection speed is improved significantly.

Our pedestrian classifier is trained by combining HOG feature with linear SVM. With the ROIs, the performance of our detector is improved significantly. Experimental results indicate that the detection algorithm is robust on the three density levels of PETS 2012 dataset.

2.2. Tracking-by-detection with an optimized observation model

Based on the data association algorithm [9], we relate no more than one detection result with a tracker. According to the detection result, we build a matching function $S(tr, d)$ for each pair (tr, d) of detection d and tracker tr. Then we associate its position and its size to get a set of function value. Finally according to Hungarian algorithm [10], we find out the most relevant detecting result of each tracker.

For each tracker we built a particle filter as [11] to predict the state of a tracker in next frame. The state includes the position and speed of a tracker. In order to track the states change of one tracker from beginning to the end, we preserve its three states, namely the Original state, Current state and Max-difference state.

We propose a new model to fuse the three states of a tracker and the associated detection result. S_O denotes the Original State, S_C denotes the Current State, and S_M denotes the Max-difference State. The states of tracker tr_i *is* $S^{(i)} = \{S_O{}^{(i)}, S_C{}^{(i)}, S_M{}^{(i)}\}$ and its observation model of tracker tr_i is:

$$p\left(y_t \middle| x_t^{(i)}\right) = \alpha * p\left(y_t \middle| x_t^{(i)}, D^{(i)}\right) + \beta * p\left(y_t \middle| x_t^{(i)}, S^{(i)}\right) \tag{3}$$

$$\begin{aligned} p\left(y_t \middle| x_t^{(i)}, S^{(i)}\right) = \\ p\left(S^{(i)} = S_O{}^{(i)}\right) * p\left(y_t \middle| x_{t-1}^{(i)}, S^{(i)} = S_O{}^{(i)}\right) + \\ p\left(S^{(i)} = S_C{}^{(i)}\right) * p\left(y_t \middle| x_{t-1}^{(i)}, S^{(i)} = S_C{}^{(i)}\right) + \\ p\left(S^{(i)} = S_M{}^{(i)}\right) * p\left(y_t \middle| x_{t-1}^{(i)}, S^{(i)} = S_M{}^{(i)}\right) \end{aligned} \tag{4}$$

where$p\left(y_t\middle|x_t^{(i)},D^{(i)}\right)$ represents the observed value from the detection result. $p\left(y_t\middle|x_t^{(i)},S^{(i)}\right)$ refers to the observed value from three states.

2.3. Three states of trackers

Original state It is the initial state of a tracker. We train a classifier using MIL algorithm once a tracker associates with a detection result. Then this classifier will not update any more. The preservation of Original State is used to track a tracker if it is occluded by another tracker after its initialization. The Original State may also make the tracking drift return to the right results.

Current state It refers to the current state of a tracker in a certain period of time. This classifier will update in each frame. When the tracker is occluded, the classifier of Current State will initialize and preserve the features of the tracker being occluded. When the occlusion is end, the classifier will be initialized again. The Current State can deal with great changes of tracker in occlusion, and continue tracking when the occlusion is end.

Max-difference state It is the maximum difference state, which refers to the state of the tracker which is the most different one from other trackers. The initialization method is the same with the method in original state. The function that defines difference is:

$$T = C(tr) - \frac{\sum_{i=0}^{num} C(tr_i)}{num} \tag{5}$$

Here, T is the value of difference and its range is [-1, 1]. $C(tr)$ is trained for Max-difference State of the tracker and its range is [0, 1]. tr_i refers to a tracker which is around the current tracker tr. num is the number of trackers around tracker tr. We define a circular area around tracker tr and its radius is 2.5 times of the height of tracker tr. If one tracker is in this area, we regard it as an around tracker of tr. We only update this state classifier when T is greater than a particular threshold value.

2.4. Optimizing the observation model

Detection term For the term associated with detection result, we get:

$$p\left(y_t\middle|x_t^{(i)},D^{(i)}\right) = \delta(\,tr_i) * p_{\mathcal{N}}(p - d^*) \tag{6}$$

where $p_{\mathcal{N}}(p - d^*)$ means the normal distribution of the distance between each particle p of tracker tr_i and its associated detection d^*. $\delta(\,tr_i)$ is an indicator function,. When the tracker has an associated detection result, it returns 1 and 0 otherwise. In this way, when the tracker finds an associated result, it will strongly guide the particle of the tracker.

Fusing multiple states term For the values of $p\left(S^{(i)} = {S_O}^{(i)}\right)$, $p\left(S^{(i)} = {S_C}^{(i)}\right)$ and $p\left(S^{(i)} = {S_M}^{(i)}\right)$, we take into account under two different circumstances:

When tracker tr_i is occluded, we always believe Current State. However, the Original State and Max-difference of the tracker may lose the significance because of being occluded by other trackers. Therefore, we got the results below:

$$p\left(S^{(i)} = {S_O}^{(i)}\right) = 0, \quad p\left(S^{(i)} = {S_C}^{(i)}\right) = 1, \quad p\left(S^{(i)} = {S_M}^{(i)}\right) = 0. \tag{7}$$

When the tracker is not occluded, we expect that when $p\left(y_t\middle|x_t^{(i)},S^{(i)}\right)$ is at maximum, the E_S which represents the difference between the features of tracker tr_i and its around trackers will be at maximum:

$$E_S = \left[log\frac{p\left(y_t\middle|x_{t-1}^{(i)},S^{(i)}\right)}{\sum_{j=0}^{num} p\left(y_t\middle|x_{t-1}^{(j)},S^{(i)}\right)/_{num}}\right] \tag{8}$$

num refers to the number of around trackers. E_S should be at maximum as well. Then the optimization problem can be reformulated:

$$\begin{aligned} &max\ \left(p\left(y_t\middle|x_t^{(i)},S^{(i)}\right) * E_S\right), \\ s.t.\quad &0 \le p\left(S^{(i)} = {S_O}^{(i)}\right), p\left(S^{(i)} = {S_C}^{(i)}\right), \\ &p\left(S^{(i)} = {S_M}^{(i)}\right) \le 1, \\ &p\left(S^{(i)} = {S_O}^{(i)}\right) + p\left(S^{(i)} = {S_C}^{(i)}\right) + p\left(S^{(i)} = {S_M}^{(i)}\right) = 1. \end{aligned} \tag{9}$$

And the values of $p\left(S^{(i)} = {S_O}^{(i)}\right)$, $p\left(S^{(i)} = {S_C}^{(i)}\right)$ and $p\left(S^{(i)} = {S_M}^{(i)}\right)$ are educed by exterior point penalty function method.

3. Multi-View Detection and Tracking

In our system, the monitored area is divided into a grid of n locations $\{x_1, x_2, \ldots\ldots x_n\}$. For a given location i on the ground plane, a rectangle of the motion blob which means a pedestrian standing at location i would produce, we name this rectangle r_i^k.

$x = \{x_1, \ldots, x_n\}$	The locations on ground plane
n	The number of locations
X_i	The Boolean random variable standing for the presence of an pedestrian at location *i*.
r_i^k	The rectangle corresponding to location i in camera *k*
R_i^k	The Boolean random variable standing for the presence of an pedestrian at r_i^k from view *k*
$an_k(R_i^k)$	The set of ancestor nodes of R_i^k in SBN *k*.
$\mathrm{C} = \{C_1, \ldots, C_K\}$	The set of cameras we used in our system
K	The number of cameras in our system
$\delta = \{\delta_1, \ldots, \delta_n\}$	The set of SSNS for all locations
$\mathrm{D} = \{D_1, \ldots, D_k\}$	The set of images from background subtraction in all camera

$\|D_k\|$	The number of pixels in D_k .
$(w,h)_k$	The pixel whose image coordinate is (w,h) in camera k.

3.1. Multi-view Detection by Tracking

We estimate the probabilities of each location on ground plane occupied by pedestrians using the MBN (multi-view Bayesian Network) model, which is composed by some Single-view Bayesian Networks (SBNs).

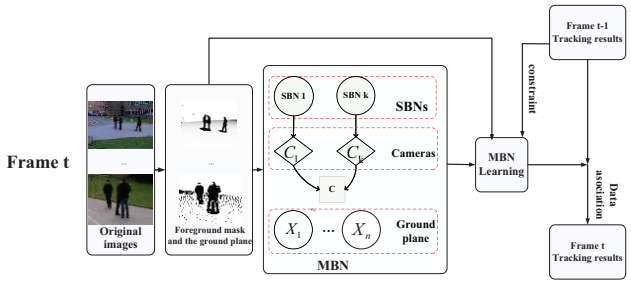

Fig 1 Framework of multi-view detection-and tracking system.

Single-view Bayesian network (SBN)

We built a single camera Bayesian network (SBN) for each camera to indicate potential occlusion relationship in single view. A SBN is organized as follow:

1.R_i^k is the ith node in SBN k.

2. R_j^k is a parent node of R_i^k in SBN k, if r_j^k occlude the r_i^k in camera k (include part occlusion).

Figure 1 shows an example of SBN. Based on the SBN:

$$P(R_i^k)=\sum_{\{\tau_j^k\in\{0,1\}|R_j^k\in an_k(R_i^k)\}} P\left(R_i^k\mid\{R_j^k=\tau_j^k\}_{R_j^k\in an_k(R_i^k)}\right)P\left(\{R_j^k=\tau_j^k\}_{R_j^k\in an_k(R_i^k)}\right); \quad (10)$$

Here we estimate occupancy possibility through estimating the part where is not occluded by other pedestrians:

$$P\left(R_i^k\mid\{R_j^k=\tau_j^k\}_{R_j^k\in an_k(R_i^k)}\right)=\frac{\sum_{(w,h)_k\in r_i^k}\left(\prod_{R_j^k\in an_k(R_i^k)\cap(w,h)_k\in r_j^k}\left(1-\tau_j^k\right)\right)}{\sum_{(w,h)_k\in r_i^k}1}; \quad (11)$$

Multi-views Bayesian Network (MBN)

In order to estimate the occupancy possibility $\mathrm{P}(X_i)$, we integrate all SBNs together as a multi-view Bayesian network (MBN).

$$P(X_i)=\sum_{k=1}^{K}P(C_k)P(X_i\mid C_k)=\sum_{k=1}^{K}P(C_k)P\left(R_i^k\right) \quad (12)$$

where $\mathrm{P}(C_k)$ is the weight value of camera k. In order to cope with the NP-hard problem of computing Eq.(11), we introduce $\delta=\{\delta_1,\dots,\delta_n\}$ named "subjective supposing" nodes state (SSNS) for MBN, where δ_i is the Boolean variable indicates whether the location x_i is occupied by a pedestrian. There are some Bayesian properties:

1) δ_j and R_i^k $(\mathrm{k}=1,2\dots\mathrm{n};\mathrm{i}\neq\mathrm{j})$ are independent with each other.

2) δ_i is the stationary state of X_i and R_i^k, therefore:

$$P\left(R_i^k=\delta_i|\delta_i\right)=1;P\left(R_i^k\neq\delta_i|\delta_i\right)=0 \quad (13)$$

Hence we get this:

$$P(X_i=1|\delta)=\delta_i\sum_{k=1}^{K}P(C_k)P\left(R_i^k\mid\{R_j^k=\delta_j\}_{R_j^k\in an_k(R_i^k)}\right); \quad (14)$$

Using Eq.(11) and Eq.(14), we can estimate occupancy possibility for each location if we know SSNS.

MBN learning using tracking information

Suppose that the pixel $(w,h)_k$ belongs to $\{r_j^k\}$. In the ideal situation, foreground pixels all come from the pedestrians and pedestrians only appear in the foreground area in each view. With this assumption, we define the loss function $\Psi(w,h)_k$ to each foreground pixel $(w,h)_k$ in camera k, which quantifies the difference between the final results estimated from (14) and ideal situation to pixel$(w,h)_k$.

$$\Psi\left((w,h)_k|\delta\right)=\begin{cases}\prod_{\mathrm{j}}(1-\mathrm{P}(X_j=1|\delta));if\,(w,h)_k\in foreground\\ 1-\prod_{\mathrm{j}}(1-\mathrm{P}(X_j=1|\delta));if\,(w,h)_k\in background\end{cases} \quad (15)$$

Given the occupancy probability of all locations estimated from (14), the conditional probability $P(D_k|\mathrm{X},\delta)$ will be:

$$\mathrm{P}\left(D_k|\mathrm{X},\delta\right)=\exp\left(\frac{-\sum_{(w,h)_k}\gamma(w,h)_k\Psi\left((w,h)_k|\delta\right)}{|D_k|}\right) \quad (16)$$

$\gamma(w,h)_k$is the weight value of pixel $(w,h)_k$,it is different only between foreground and background pixels in our experiment. We suppose various background subtraction images from different views are independent from each other, so the likelihood function is defined as follow:

$$F(\delta_1,\dots\dots\delta_n)=\ln\left(P\left(D\mid X,\delta\right)\right)=\sum_{k=1}^{K}\ln(P\left(D_k\mid X,\delta\right)); \quad (17)$$

where $\forall i,\delta_i=\{0,1\}$. Optimizing the likelihood function:

$$\left(\delta_1,\dots\dots\delta_n\right)=\arg\max F(\delta_1,\dots\dots\delta_n); \quad (18)$$

In order to solve this problem, we relax SSNS to the continuous domain. Typically, $\varepsilon_i\in\{-\infty,+\infty\}^n$ is used to replace $\delta_i\in\{0,1\}$ with the sigmoid function, then we could obtain that:

$$\left(\varepsilon_1,\dots\dots\varepsilon_n\right)=\arg\max F(\varepsilon_1,\dots\dots\varepsilon_n); \quad (20)$$

To the locations which are located in the area where a new tracker occurs and disappears impossibly:

1. If $\forall tr_j\epsilon Tr\ \left|x_i-tr_j\right|>maxdistance,\ \varepsilon_i=0;$
2. $\sum_i\varepsilon_i>|Tr|-\theta$ where Tr are the trackers which disappear impossibility in this frame. θ is a constraint which is introduced to relax the constraint.

After optimizing (20) with this two constraints, we put ε into (11) and (14), then we get the final detection

result $\mathrm{P}(X_i = 1)$.

3.2. Multi-view tracking by detection

We employ the data association method [10] to assign at most one location to at most one tracker. Our data association method evaluates a matching matrix C between locations on the ground plane and the trackers.

$$\mathrm{C}(i,j) = S(tr_i, x_j) \tag{21}$$

$S(tr_i, x_j)$ is a matching function for the j th tracker and the i th location pair. The higher the score is, the better the match is. In our method, we calculate $S(tr_i, x_j)$ as follows:

$$S(tr_i, x_j) = P(\mathrm{X}_j = 1) g(tr_i, \mathrm{x}_j) \sum_{k=1}^{K} w_k S_k(tr_i^k, r_j^k) \tag{22}$$

where $P(X_j = 1)$ is the occupancy possibility for location j through (14). w_k is the weight value to camera k. $S_k(tr_i^k, r_j^k)$ is got by the monocular tracking method [12].

$$g(tr_i, \mathrm{x}_j) = p_N(|tr_i - \mathrm{x}_j|); \tag{23}$$

Where $g(tr_i, \mathrm{x}_j)$ denotes the normal distribution evaluated for the distance between tracker and the location.

4. Event Recognition

Absolute and relative orientations and magnitudes of velocities are crucial in analyzing these events. So we use both of them in our approach to build the feature.

First, the features used in [13] are utilized to select features in each frame that can be tracked well and correspond to physical points in the world. After that these features' locations are further refined into sub-pixels. Then the pyramid Lucas & Kanade method [14] is employed to calculate optical flow for every point at every pyramid level. Based on the optical flow of feature points, histograms of orientations and magnitudes are computed in 2-D rectangular to describe the absolute motion information of the pedestrians. And histograms computed in polar coordinate are for the relative motion information, see Fig.2. Then we concatenate them together and use the SVM to classify each class (e.g. walking vs. not walking).

5. Experiments

5.1. Single View Detection and Tracking Results

The parameters of our single view detection and tracking task are set as below: 1) The number of depth levels is set as 30. 2) The initial sample positions of trackers are drawn from a Normal distribution ($\sigma = 16$). 3) The size of a tracker is set to the average of the last seven associated detections. 4) The number of particles for each tracker is 150. 5) The ratio is about 3: 2 between detection term and multiple states term.

Due to the use of ROIs based on geometric constraints, our average detection speed is increased from 0.6 fps to 10 fps. The compare results with state-of-art are shown in Table 1, 2 and Fig 3 which indicates that our method could achieve much better performance due to the use of detection with geometric constraints based ROIs and tracking with our optimized observation model.

Table 1 Detection results in single view (threshold = 0.5)

Task	Method	MODA	MODP
S2.L1 view1	Ours	0.96	0.81
	Yang[15]	0.95	0.55
	Breitenstein[16]	0.89	0.60

Table 2: Tracking results in single view (threshold = 0.5).

Task	Method	MOTA	MOTP
S2.L1 view1	Ours	0.96	0.80
	Yang[15]	0.95	0.55
S2.L2 view1	Ours	0.60	0.60
	Breitenstein[16]	0.50	0.51
S2.L2 view2	Ours	0.66	0.70
S2.L3 view1	Ours	0.87	0.54
	Breitenstein[16]	0.68	0.52
S2.L3 view2	Ours	0.73	0.68

5.2. Multi-view Detection and Tracking Results

The parameters of our multi-view detection and tracking task are set as below: 1) the weight value of foreground and foreground pixels are respectively equal to 2 and 1. 2) $maxdistance = 2\mathrm{m}$ and $\theta = 0.5$.

The compare results shown in Table 2 indicate that our MBN could effectively reduce the "phantom" phenomena. The results of Berclaz and Ge are reported in [17].

Table 3 Tracking results with multi cameras (threshold = 0.5).

Task	Method	MODA	MODP	MOTA	MOTP
S2.L1 view1	Ours	0.95	0.76	0.95	0.76
S2.L1 view5	Ours	0.86	0.72	0.84	0.70
S2L1. view6	Ours	0.91	0.68	0.90	0.67
Median view values	Ours	0.91	0.72	0.90	0.70
	Berclaz	0.76	0.62	0.75	0.62
	Ge	0.85	0.45	0.75	0.46

5.3. Event Recognition Results

The dataset is split into training (75%) and test (25%) sets. We report the classification accuracy on test set for View 1.

Table 4. Videos used for crowd event detection.

class	set: video[frames]
walking	S3:14-16[0-35,108-162];S3:14-31[0-end]; S3:14-33[294-310]
running	S3:14-16[36-107,163-end];S3:14-33[328-377]
evacuation	S3:14-33[328-377]
dispersion	S3:14-27[96-144, 270-318]
merging	S3:14-33[0-180]
splitting	S3:14-31[48-130]; S3:14-33[328-377]

Table 5. Classification accuracy on test set for View 1.

Event	Wal.	Run.	Eva.	Dis.	Mer.	Spl.

Accuracy	0.97	0.98	0.99	0.94	0.99	0.98

6. Conclusions

In this paper, we focus on the task of single view and multi-view detection, tracking and crowd events recognition. For the purpose of eliminating the negative influence of clutter environment, geometric constraints are utilized to generate ROIs. Combined with the ROIs, detection accuracy and speed of our HOG based detector are increased significantly in single view detection task. Then, an optimized observation model is utilized to address the ID switching or tracking drifting problem in single view tracking task. For the multi-view detection task, we introduce the multi-view Bayesian network (MBN) to reduce the "phantom" phenomena which frequently happen in the general multi-view detection tasks. At last, a motion-based event recognition method is proposed to handle the event recognition task. Experimental results for the three PETS 2012 tasks indicate that our detection, tracking and event recognition method are very promising.

References

[1] D. Mitzel, P. Sudowe, and B. Leibe. Real-Time Multi-PersonTracking with Time-Constrained Detection. In BMVC, 2011.

[2] A. Ess, B. Leibe, L.V. Gool. Depth and Appearance for Mobile Scene Analysis. In ICCV, 2007

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[4] Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV,*2007.

[5] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007.

[6] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[7] B. Babenko, M. Yang, S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[8] B. Babenko, M. Yang, S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[9] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV,* 2009.

[10] R. Hess and A. Fern. Discriminatively Trained Particle Filters for Complex Multi-Object Tracking. In *CVPR*, 2009.

[11] H. Kuhn. The hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2:83–87, 1955.

[12] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. PAMI, 33(9), 2011.

[13] J. Shi and C. Tomasi. Good features to track. (CVPR) 593-600, 1994

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, 121—130, 1981

[15] J. Yang, Z. Shi, P. Vela, and J. Teizer. Probabilistic multiple people tracking through complex situations. In IEEE Workshop Performance Evaluation of Tracking and Surveillance, 2009.

[16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. PAMI, 33(9), 2011.

[17] A. Ellis and J. Ferryman. Pets2010 and pets2009 evaluation of results using individual ground truth single views. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS), pages 135 – 142, 2010.

Fig 2 Features are extracted in different coordinate systems. (a) Histograms of orientation and magnitude are computed in 2-D rectangular coordinate system. (b) Histograms of relative orientation (α) and relative magnitude (v) (referring to O) are computed in polar coordinate system.

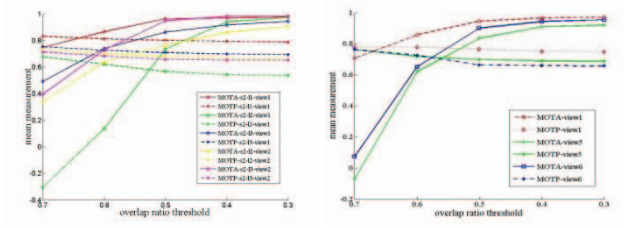

Fig 3 Result of single view and multiple tracking system on different data.

Fig 4 Exemplary single view tracking results for the PETS 2012 tracking datasets

Fig 5 Some results of our multi-view tracker on the PETS 2012 dataset. The right-most column represents the corresponding detections on the ground plane.