Multimed Tools Appl (2015) 74:729–742
DOI 10.1007/s11042-014-2177-x

Multiple human detection and tracking based on head detection for real-time video surveillance

Ruiyue Xu · Yepeng Guan · Yizhen Huang

Received: 4 November 2013 / Revised: 28 April 2014 / Accepted: 23 June 2014 / Published online: 5 October 2014

Abstract Multiple human detection and tracking is a very important and active research topic in computer vision. At present, the recognition performance is not satisfactory, which is mainly due to the fact that the full-body of human cannot be captured efficiently by cameras. In this paper, an improved method is developed to detect and track multiple heads by considering them as rigid body parts. The appearance model of human heads is updated according to fusion of color histogram and oriented gradients. An associative mechanism of detection and tracking has been developed to recover transient missed detections and suppress transient false detections. The object identity can be kept invariant during tracking even if unavoidable occlusion occurs. Besides, the proposed method is fast to detect and track multiple human in a dynamic scene without any hypothesis for the scenario contents in advance. Comparisons with state-of-the-arts have indicated the superiority and good performance of the proposed method.

Keywords Human detection and tracking · Real-time · Associative mechanism

1 Introduction

Intelligent visual surveillance has been gaining more attention from the community due to the increasing importance and needs of crime prevention and anti-terrorist applications. Human detection and tracking from video sequences is naturally a key issue for intelligent visual surveillance. Many methods have been proposed for human detection and tracking so far: Zhao et al. [20] integrated fast gradient Hough transform, hair-color distribution model and circle existence model to detect human heads. Yang et al. [18] used skin color and head shape model to achieve this task. The color used in the method mentioned above may be confused with other objects in complex background. Besides, the above method is not suitable for complex situations such as several people moving in the scene with partial occlusion. Yuk et al. [19] proposed a probabilistic model based shape contour matching algorithm to detect

R. Xu · Y. Guan (✉) · Y. Huang
School of Communication and Information Engineering, Shanghai University, Shanghai, China
e-mail: ypguan@shu.edu.cn

Y. Guan
Key Laboratory of Advanced Displays and System Application, Ministry of Education, Shanghai, China

and locate head-like objects. For this method, good motion segmentation or edge detection result is needed as an initialization step. Besides, they assumed that the head-like shape is an ellipse or a special shape. Xie et al. [16] used an adaptive detector to detect heads based on the Histogram of Gradients (HoG) feature. It is time-consuming to compute HoG feature descriptor and a large head region is needed to extract HoG feature, which limits its practical application. Wang and Tian [14] detected head based on head detectors with Haar feature and Adaboost algorithm. The Haar feature is robust in complex dynamic scenes and relatively less time-consuming compared with HoG feature. The main challenge in using a detector for head tracking is that it is prone to make errors when the detector's output is unreliable. The increasing of detection rates would result in the increasing of false positive rates, so a tradeoff between the detection rates and false positive rates should be determined. To alleviate this dilemma, head plane estimation based on 3D information was employed in [1]. The head plane estimation makes a few key assumes, for example, the camera's intrinsic parameters need to be known beforehand, and all human heads are approximately the same size. Moreover, the 3D head plane refinement increases the complexity of the algorithm.

To address these issues, many researchers have proposed tracking methods that utilize object detection [3, 4, 6, 15, 17]. These tracking methods link detection responses to trajectories by global optimization based on position, size, and appearance similarity. They are with high complexity and are prone to yield identity-switches and trajectory fragments due to false detections and occlusion. In [2], Benfold et al. presented a multi-target tracking system that is designed specifically for stable and accurate head location estimates. They used data association over a sliding window of frames, and their system is multi-threaded combining asynchronous HoG detections with simultaneous KLT tracking and Markov-Chain Monte-Carlo Data Association (MCMCDA). Their system is different from our proposed method mainly in the following three parts: 1) We used foreground segmentation to speed up the detection and reduce the false alarm rate; 2) We applied a particle filter to track each object while Benfold et al. using simultaneous KLT tracking; 3) We used AdaBoost cascade detectors for real-time head detection.

Aiming at the limits aforementioned, a novel framework has been proposed in this paper to detect and track multiple humans for real-time general purpose video surveillance: A multi-scale wavelet transformation using frame difference is developed to segment motion foreground. AdaBoost cascade classifier is adopted to detect heads only over Region Of Interest (ROI) specified by the extracted foreground bounding rectangle instead of the entire input image. One particle filter is employed for each head during tracking. A new trajectory for an object is initialized in the subsequent detection. A head-confirmation mechanism is proposed to confirm the head and recover the missing head not being detected. We keep the object identity invariant during tracking even if unavoidable occlusion occurs. The first contribution of the paper is that, one particle filter is employed for each tracked head and the appearance models of heads are updated based on fusion of color histogram and oriented gradients. The object identities are kept invariant during tracking even if unavoidable occlusion occurs in dynamic scenarios. A second contribution is that an associative mechanism of detection and tracking has been developed. Some missed detections can be recovered and some false detections can be eliminated from the proposed mechanism. Besides, the result of detection can be used to correct tracking result to improve the accuracy. A third important contribution is to fast detect and track multiple human in ordinary hardware settings from a general scene without any hypothesis for the scenario contents beforehand. Comparisons with state-of-the-arts have indicated the superior performance of our method.

The rest of the paper is organized as follows: Head detection is described in Section 2; In Section 3, head tracking is discussed; Head confirmation is presented in Section 4;

Experimental results and analysis are shown in Section 5 and followed by some conclusions in Section 6.

2 Head detection

2.1 Foreground segmentation

In order to reduce head detection time and eliminate false head detection from background, multi-scale wavelet transformation (WT) using frame difference is developed to segment foreground. The low-pass and high-pass filters of the WT naturally break a signal into similar (low-pass) and discontinuous (high-pass) sub-signals [7], which effectively combines the two basic properties into a single approach. The low-pass sub-signals serve as a similar functionality as the Gaussian mixture model (Gaussian filters are low-pass), while the high-pass sub-signals we used are the Sobel edge detectors (as in [7]) which serve as the data consistency term of adjacent pixels in the segmentation algorithm. Since the HSV color space corresponds closely to the human perception of colors and it explicitly separates chromaticity and luminosity, it is selected to be used here instead of other color models. We define a foreground mask P_f for each pixel (x, y) as follows.

$$P_f = \begin{cases} 1, E_{\Delta V} \geq T_{\Delta V} \wedge E_{\Delta S} \geq T_{\Delta S} \\ 0, otherwise \end{cases} \tag{1}$$

where ΔV and ΔS are the difference between the two successive frames of the value and saturation component, respectively; $E_{\Delta V}$, $E_{\Delta S}$ are multi-scale WT across ΔV, and ΔS, respectively; $T_{\Delta V}$,$T_{\Delta S}$ represent a threshold value of ΔV, and ΔS, respectively.

To remove ghost effects in which the extracted foreground region is larger than the actual moving object, the WT-based edge detection is used to extract edges of current frame as following

$$P_e = \begin{cases} 1, E_V \geq T_V \\ 0, otherwise \end{cases} \tag{2}$$

where V is the value component of current frame, T_V is a threshold value for E_v.

A logical AND operation is applied on the P_f and P_e to extract the foreground region mask P for each pixel (x, y) as follow:

$$P = P_f \wedge P_e \tag{3}$$

A morphological process can be applied to the extracted foreground for its completeness. Some examples of *CAVIAR* sequences from [12] are given in Fig. 1.

2.2 Head training and detection

Based on the foreground segmented, we detect heads only over the ROI, which is specified by the extracted foreground bounding with a rectangle (seen from Fig. 1b), instead of the entire image. Before detecting heads, we train a classifier to get a detector off line as follows. We collected 7100 images of human heads and 1, 2000 negative images without heads from different videos and cropped them to the size of 12×12 pixels [1] at first. Some of these images are collected using our own

[1] As an improvement over [1], our system is able to detect human heads with the minimum size 12x12, which is used for the special application of long-distance video surveillance.

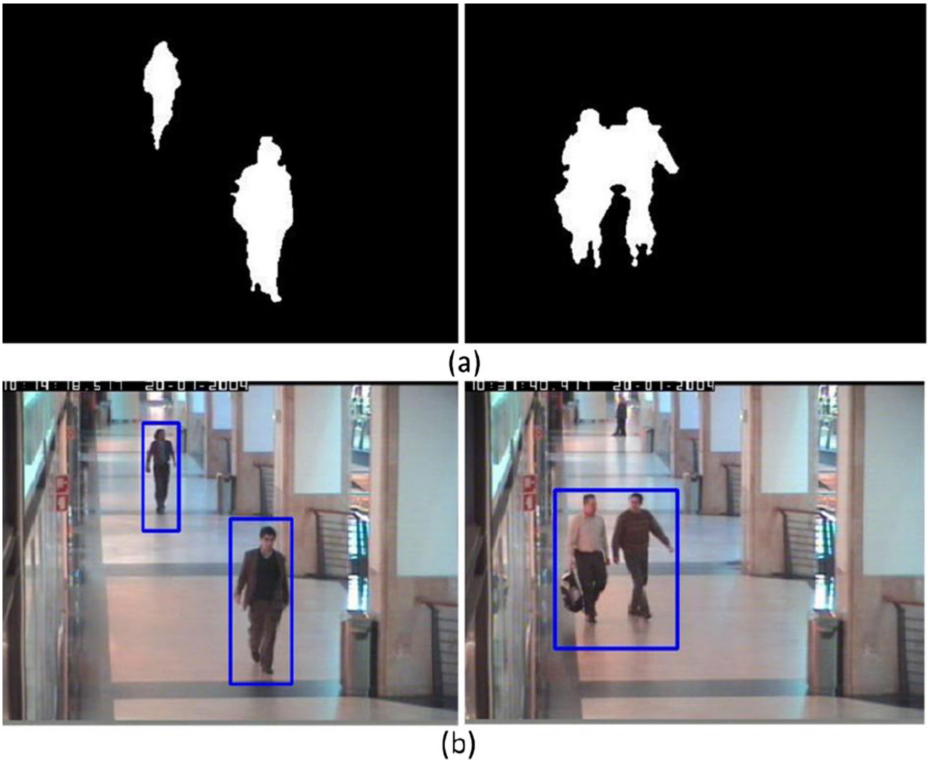


Fig. 1 Motion object extraction results. **a**. An example binary foreground mask. **b**. Foreground objects detected with bounding *rectangles*

video cameras while the rest are from the Internet. We train these samples using the AdaBoost cascade detector [13]. We detect heads based on the trained detector at an interval of one frame. Some examples of detection results are given in Fig. 2.

3 Head tracking

One particle filter [5] is employed for each tracked head, namely, each head trajectory contains a particle filter. The initial distribution of the particle is centered at the location of the head as detected and the initial weight of each particle is set $w_0=1/30$ in the paper. [1] also proposed to use particle filters and an appearance model for head tracking, however, in this paper, we adopted a Gaussian kernel to compute the particle's likelihood, and the confirmation-by-classification mechanism in [1] is discarded.

4 Head confirmation

Head confirmation consists of four functions including false detection suppression, recovery from miss, data association of detection and tracking, and occlusion handing. Some details are described as follows.

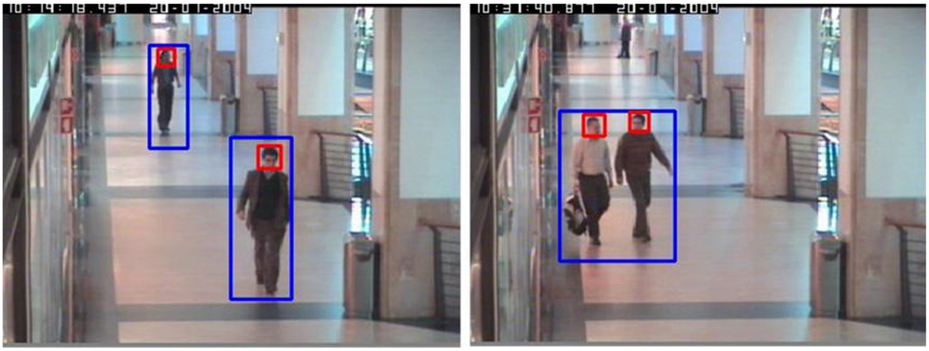


Fig. 2 Some results of heads detection

4.1 False detection suppression

In the process of detecting head as mentioned above, some tracking errors may exist due to false detection. To overcome this problem, a head count method is developed as follows. When a head is detected, we do not consider it as a true head directly at first. We allocate a transient trajectory for it. The initial value of head count O_j with trajectory j is set to be 0.5. For the updated head's position in each subsequent frame, we confirm the estimated position through detection. When the trajectory j is confirmed, we increase the head count O_j; otherwise, decrease it. Besides, we set both threshold values as O_{up1} and O_{down}. The initial head of the trajectory is confirmed when the head count O_j first reaches to the value O_{up1}, and eliminated when the head count O_j downs to the value O_{down}.

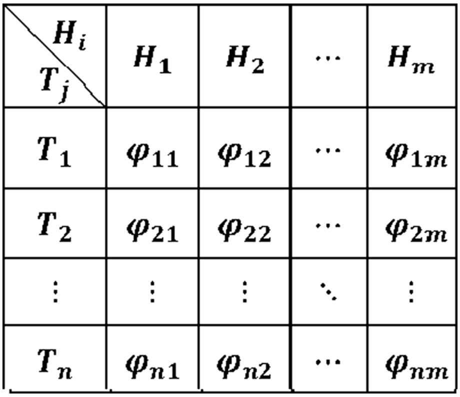
4.2 Recovery from miss

Since head may be missed in the process of detecting and tracking caused by occlusion or incomplete foreground segmented, it is necessary to recover true detection from miss. To address this issue, new heads not being predicted previously by the existing trajectories over the ROI are searched. Discriminating the detected heads about whether they are new ones or previous predicted ones by existing trajectories based on data association of detection and tracking will be discussed later. Since some true motion trajectories of head may not be detected for several frames, both head count update strategies are employed to solve this problem as follows.

$$O_j = \begin{cases} O_j - 0.5, O_j \geq O_{down} \\ remove, O_j < O_{down} \end{cases} \tag{4}$$

$$O_j = \begin{cases} O_j + 0.5, 0 \leq O_j < O_{up1} \\ O_{jm}, O_{up1} \leq O_j < O_{jm} \\ O_j + 1, O_{jm} \leq O_j < O_{up2} \\ O_{up2}, O_j \geq O_{up2} \end{cases} \tag{5}$$

where O_{down}, O_{up1}, O_{jm}, O_{up2} are thresholds, *remove* means that the trajectory should be considered as a false one and be removed in the next frame; O_j in Eq.(4) represents the case of estimated head position of trajectory j being not confirmed at the current frame, while O_j in Eq.(5) represents the case of estimated head position of trajectory being confirmed at the current frame (discussed in experimental results further).



H_i / T_j	H_1	H_2	$\cdots$	H_m
T_1	φ_{11}	φ_{12}	$\cdots$	φ_{1m}
T_2	φ_{21}	φ_{22}	$\cdots$	φ_{2m}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
T_n	φ_{n1}	φ_{n2}	$\cdots$	φ_{nm}

Fig. 3 An incidence matrix *R*

4.3 Data association of detection and tracking

We introduce an incidence matrix based on minimum Euclidean distance to construct associative mechanism of detection and tracking in this section. Denote some estimated head positions for each existing trajectory as T_j, $T_j \in \{T_1,T_2,\ldots,T_n\}$ (n is the number of existing trajectories), and the detected head positions at the current frame as H_i, $H_i \in \{H_1,H_2,\ldots,H_m\}$ (m is the number of detected heads at the current frame). The incidence matrix R is defined as following (Fig. 3).

For each detected head H_i, $i \in \{1,2,\ldots,m\}$, we calculate the Euclidean distance d_{ji} between H_i and T_j, $j=1,\ldots,n$ separately. Find a minimum distance $d_{ji}(min)$ and its corresponding T_p, $p\in\{1,2,\ldots,n\}$. Set the correspondingφ_{pi} ($j=p$) as 1, and othersφ_{ji} ($j\neq p$) as 0. There is only one value 1 in each column at the matrix R, while there may be more than one value 1 for each row at the matrix R. The formation of this matrix is of similar rationale with the well-known inter-pixel correlation model introduced in [9].

We realize the associative mechanism of detection and tracking based on the matrix R as follows. For each row at the matrix R, calculate the sum as following.

$$r_j = \sum\nolimits_{i=1}^{m} \varphi_{jm} \tag{6}$$

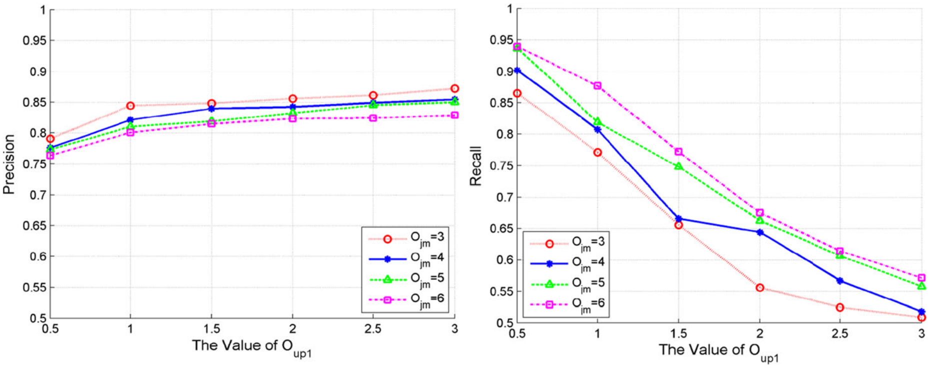


Fig. 4 *Precision* and *Recall* with different values of O_{up1} and O_{jm} for the CAVIAR dataset

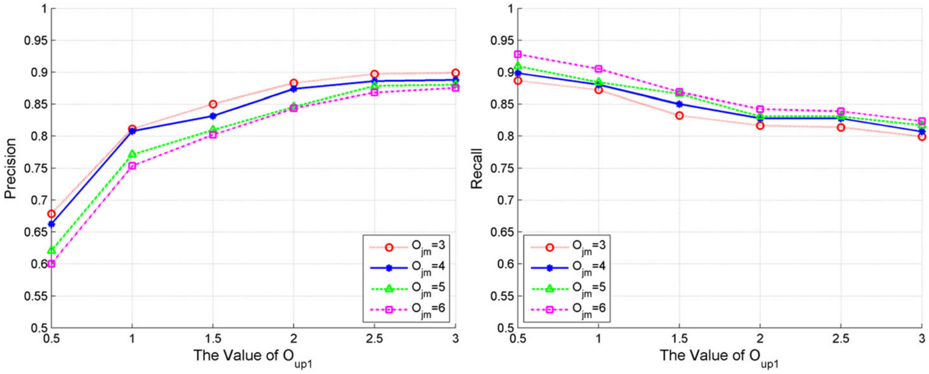


Fig. 5 *Precision* and *Recall* with different values of O_{up1} and O_{jm} for the UT-Interaction dataset

There are three cases for r_j as follows. (1) When r_j=0, it indicates that no detected head is associated with the trajectory *j*. In this case, the tracking result is the final one. The trajectory *j* does not need to be corrected. (2) When r_j=1, it indicates that there is only one head is associated with the trajectory *j*. In this case, we need to judge whether this corresponding head *i* is belonging to the trajectory *j* or not. We introduce a threshold *C* which is set the width of the estimated head of the trajectory. If $d_{ji}(min)<C$, we confirm the trajectory *j* is corresponding to the head *i*. Otherwise, we consider this head *i* as a new detected head and allocate a new trajectory for it. (3) When r_j>1, it indicates that there are more than one heads are associated with the trajectory *j*. In this case, we use the size, color, and oriented gradient histogram of head to decide which head corresponds to the trajectory *j*. Interestingly, the selection process here can be explained by the optimal recovery theory [8] to some extend.

5 Experimental results and analysis

To test performance of the proposed approach, we have done experiments on a large number of real-world video sequences. To evaluate its performance with that of state-of-the-art methods, we select some public and collected by ourselves video datasets to test their performance at the

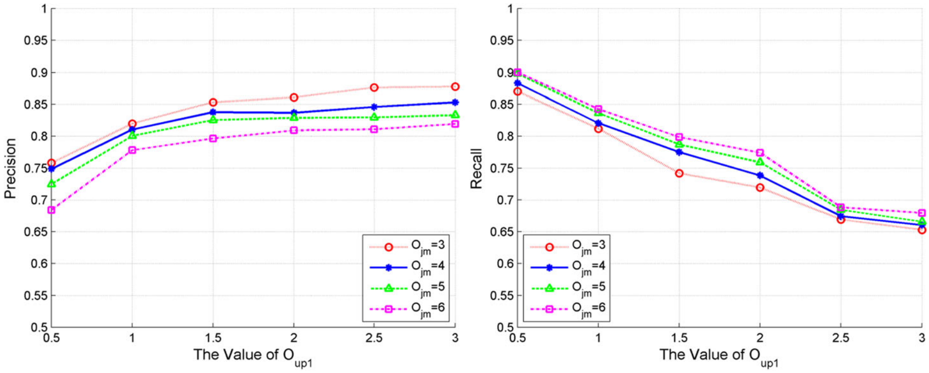


Fig. 6 *Precision* and *Recall* with different values of O_{up1} and O_{jm} for ourselves dataset

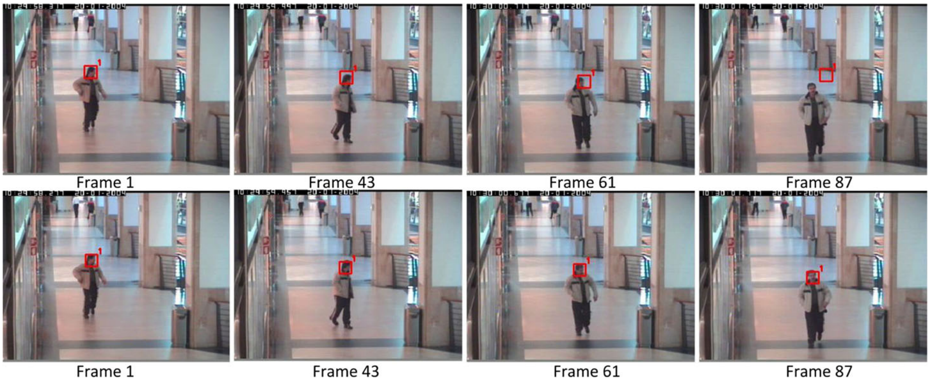


Fig. 7 Some tracking results for the *CAVIAR* dataset. Results in the first row are from [10], results in the second *row* are our results

same situations. All the experiments are performed in C++ project with OpenCV library, Pentium(R) 2.6GHz CPU and 3G RAM memory.

5.1 Tested videos

The first selected dataset is *CAVIAR* from [12], which is captured in a corridor with resolution 384×288 pixels. In the *CAVIAR* dataset, the color of human's head is similar with that of background. There are inter-object occlusions and frequent interactions between human in the dynamic scenes, which makes it difficult to detect and track multiple heads.

The second selected dataset is UT-Interaction dataset from [11], which contains 6 classes of human-human interaction, hand shaking, punching, pushing, hugging, kicking and pointing with resolution 780×480 pixels. Several pairs of interacting persons execute activities simultaneously in the scene. Each video has different background, scale and illumination. The motion of head is instantaneous and fast with no rule. Besides, occlusion occurs due to movement of hand.

Table 1 Evaluation metrics

Name	Definition
Crowds	The average number of heads per frame
GT	Number of ground truth trajectories.
MT(%)	Mostly tracked: percentage GT trajectories which are covered by track output for more than 80 % in length.
ML(%)	Mostly lost: percentage GT trajectories which are covered by track output for less than 20 % in length.
PT(%)	Partially tracked: percentage GT trajectories which are covered by track output for 20–80 % in length.
Frag	Fragments: number of times that a ground truth trajectory is interrupted in the tracking result.
IDS	Identity switches: number of times that two trajectories switch their identities
FAT	False alarm trajectories: number of output trajectories that do not correspond to ground truth trajectories and exist for more than 10 frames.
Speed	The average runtime per frame.

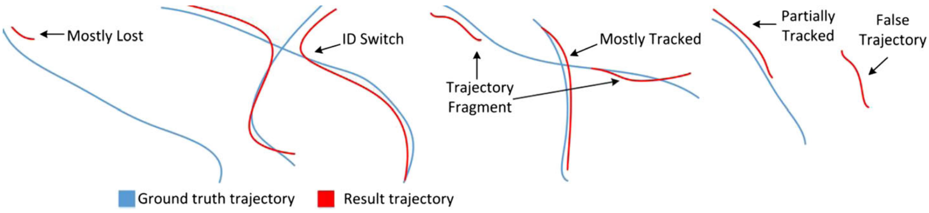


Fig. 8 Evaluation metrics

The third selected dataset is collected by ourselves captured more than 10,000 frames with resolution 400 × 304 pixels in a pedestrian street. Some pedestrians frequently go in and out the scene and the head is smaller than that of the above datasets.

5.2 Choice of parameters

In order to build a fair comparison with state-of-the-arts, some parameters mentioned above must be kept the same in the whole test. O_{down} in (8) is set 0.5, and O_{up2} in (9) is set as twice of O_{jm}. So we discuss O_{up1} and O_{jm} only in (9).

We adopt *Precision* (also called positive predictive value), and *Recall* (also known as sensitivity) to analysis the performance with some different O_{up1} and O_{jm} values. The *Precision* is the fraction of retrieved instances that are relevant, while *Recall* is the fraction of relevant instances that are retrieved. Some results of *Precision* and *Recall* with different O_{up1} and O_{jm} values for the selected three datasets are shown in Figs. 4, 5 and 6, respectively.

One can note that the *Precision* increases with the increasing of O_{up1}, while *Recall* decreases no matter which dataset is selected from Figs. 4, 5 and 6. On the contrary, *Precision* decreases, while *Recall* increases with increasing of O_{jm}. We choose $O_{up1}=1.0$ and $O_{jm}=4$, which makes a trade-off between the *Precision* and *Recall* for all the three datasets.

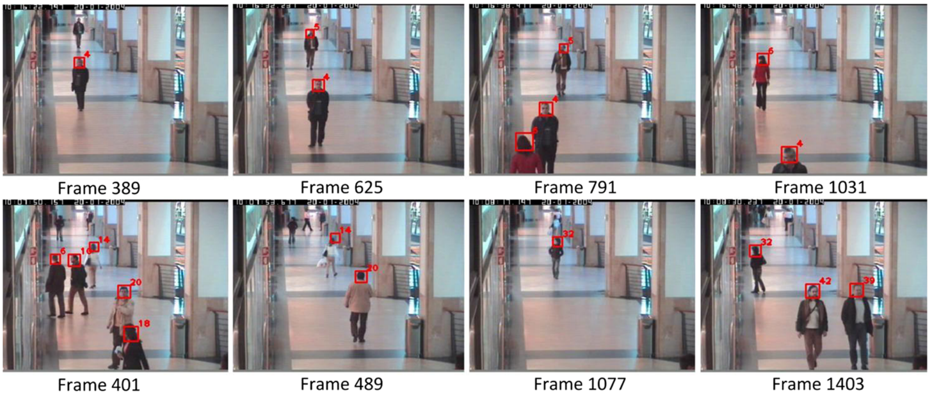
5.3 Performance evaluation with state-of-the -arts

To test whether heads can be tracked efficiently or not using the improved particle filter based on fusion of color histogram and oriented gradient one, *CAVIAR* dataset and the particle filter based on the appearance of color histogram proposed in [5] are selected. There we detect head on the first frame and track head frame by frame. Some tracking results are given in Fig. 7.

The above results show qualitative information about the effectiveness of our method in tracking heads. It is necessary to evaluate our whole performance with that of state-of-the-arts including [1, 19] quantitatively. The evaluation method proposed in [1] is used to test

Table 2 Some comparison results for CAVIAR dateset

Method	Crowd	GT	MT(%)	ML(%)	PT(%)	Frag	IDS	FAT	Speed
[3]	2.86	146	73.3	8.2	18.5	36	16	34	200 ms
[6]	2.86	146	76.7	0	23.3	31	14	41	350 ms
Proposed	2.86	146	80.8	0.7	18.5	23	10	37	180 ms



Frame 389 Frame 625 Frame 791 Frame 1031

Frame 401 Frame 489 Frame 1077 Frame 1403

Fig. 9 Some tracking results for the *CAVIAR* dataset

performance as shown in Table 1. It should be stated that our definition of *FAT* is different with the definition in [1]. The *FAT* mainly caused by two ways, one is that false detection is confirmed continuously, the other is that a true trajectory drifts for some reasons.

In order to make the definition be more visual, it is given in Fig. 8.

To establish a fair comparison, the above selected three video sequences are manually segmented to generate the ground-truth. Some results for the *CAVIAR* dataset are shown in Table 2, in which, the results of [19] and [1] are obtained by our own implemented code according to the descriptions provided by their respective authors.

From Table 2, we can find that the proposed method has the highest *MT*, the lowest values in *Frag*, and *IDS* among the three methods. Although the *FAT* is higher than that of method [19], the *ML* is much lower than that of [19]. The performance of our method is the best among the investigated methods as a whole.

In terms of the running time, the average processing time for each frame is about 180 ms for our method, while the methods in [19], [1] are 200 ms, 350 ms, respectively. It indicates that our proposed method has the fastest processing time among the investigated methods. Considering the fact that, [1] is using a much faster CPU (Intel Core i5 3.2), our method should have more advantage on this aspect.

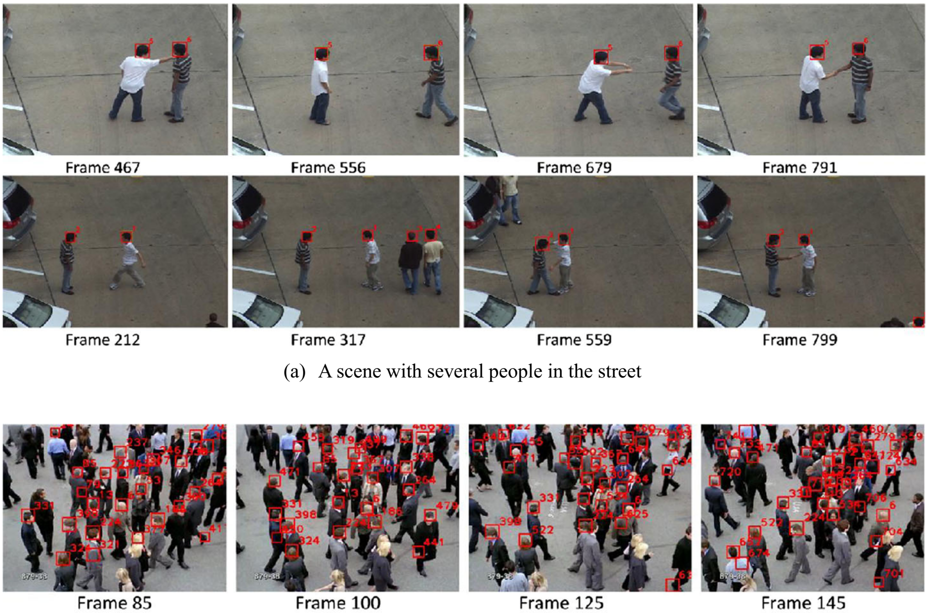
Some head tracking examples are given in Fig. 9 for the *CAVIAR* dataset.

Some results for the UT-Interaction dataset are shown in Table 3.

From Table 3, one can note that the proposed method has the highest *MT*, and the lowest *IDS* among the three methods. Although the *FAT* is higher than that of [19], the *ML* and *Frag* in our paper are much lower than that of [19]. The performance of our method is the best among the investigated methods as a whole also.

Table 3 Some results for the UT-Interaction dataset

Method	Crowd	GT	MT(%)	ML(%)	PT(%)	Frag	IDS	FAT	Speed
[3]	2.55	82	76.8	4.9	18.3	19	8	17	380 ms
[6]	2.55	82	76.7	0	22.0	15	6	23	2,400 ms
Proposed	2.55	82	82.9	1.2	15.9	16	4	19	410 ms



(a) A scene with several people in the street

(b) A scene with a large number of crowded people in the street

Fig. 10 Tracking results in UT-Interaction dataset. **a.** A scene with several people in the street, **b.** A scene with a large number of crowded people in the street

In the average running time for each frame, our method is about 410 ms, while the method in [1] is 2,400 ms. The reason for [1] with great running time is as following. The method in [1] detects heads over the whole scene frame by frame. We detect heads only over the ROI at an interval of one frame. For the method in [19], the average runtime time for each frame is 380 ms, which is similar to ours. Some head tracking examples for the UT-Interaction dataset are shown in Fig. 10.

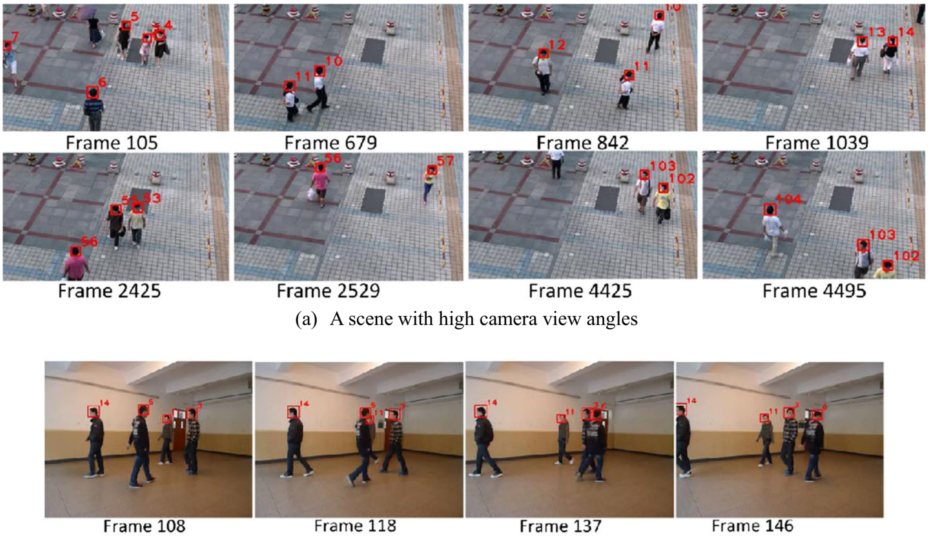
Some results for ourselves dataset are shown in Table 4. From Table 4, one can find that the proposed method has the highest *MT*, the lowest values in *Frag*, *IDS* and *FAT* among the three methods. It indicates that our method is far better than that of the investigated methods for oursevles datasets.

In the running time, the average runtime per frame in the paper is about 130 ms, while the methods in [1] is 150 ms, 230 ms, respectively. It has been shown that our system is the most excellent in terms of this performance aspect. Some examples are shown in Fig. 11 for the dataset.

To further emphasize the performance of our method in dealing with occlusion especially two or more heads overlap each other frequently, another indoor video is selected where

Table 4 Some results for ourselves dataset

Method	Crowd	GT	MT(%)	ML(%)	PT(%)	Frag	IDS	FAT	Speed
[3]	3.32	110	78.2	6.3	15.5	21	8	24	150 ms
[6]	3.32	110	81.8	0.9	17.3	18	6	28	230 ms
Proposed	3.32	110	83.6	2.7	13.7	16	5	22	130 ms



(a) A scene with high camera view angles

(b) A scene with low camera view angles

Fig. 11 Some tracking results for ours dataset. **a.** A scene with high camera view *angles*, **b**. A scene with low camera view *angles*

several people walk in a line with 27 times cross movements at a short instant (about 20 frames). It is easy to cause the IDS problem in this selected video. Some examples are shown in Fig. 12.

We test it in this video and compared it with that of methods in [19], [1], respectively. Some results are given in Table 5.

From Table 5, one can find that the *IDS* in our method are much less than that of the methods in [19] and [1]. It further indicates that the developed method has the best superiority in occlusion handling among the investigated methods.

By comparisons, it highlights that the proposed method has good performance among the investigated methods in detecting and tracking heads even if in the case of unavoidable occlusion.

6 Conclusions

An improved method is developed to detect and track multiple heads according to the fact that head is a rigid part of body. A particle filter is employed for each tracked head and head's appearance model is updated based on fusion of color histogram and oriented gradient one. An

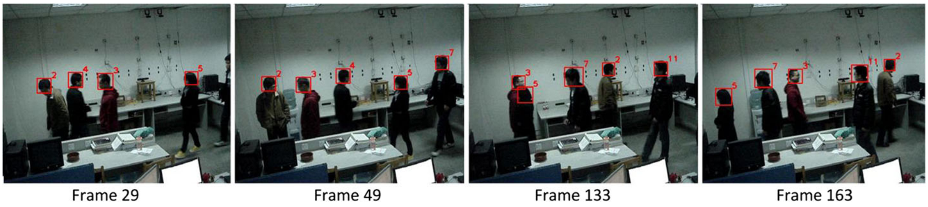


Fig. 12 Some tracking results for frequent cross-movements

Table 5 Some results for frequent cross-movements

Method	[3]	[6]	Proposed
IDS	18	14	8

associative mechanism of detection and tracking has been developed in which some missed detections can be recovered. The object identity is kept invariant during tracking even if unavoidable occlusion occurs in a dynamic scenario. Besides, the proposed method is fast to detect and track multiple human in an ordinary hardware from a general scene without any hypothesis for the scenario contents in advance. Comparative study with state-of-the-art human detection and tracking methods has indicated the superiority and good performance of our method.

In the future, we would do some robust features extraction to improve the performance of detection and tracking in dynamic scenarios.

Acknowledgments This work is supported in part by the Natural Science Foundation of China (Grant No.11176016, 60872117), and Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20123108110014).

References

1. Ali I, Dailey MN (2012) Multiple human tracking in high-density crowds. Image Vis Comput 30(12):966–977
2. Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. IEEE Comput Soc Conf Vis Pattern Recognit 1:3457–3464
3. Berclaz J, Fleuret F, Fua P (2006) Robust people tracking with global trajectory optimization. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 1:744–750
4. Breitenstein MD, Reichlin F, Leibe B, Koller-Meier E, Van Gool L (2009) Robust tracking-by-detection using a detector confidence particle filter. Proceedings of IEEE International Conference on Computer Vision, 1515–1522
5. Czyz J, Ristic B, Macq B (2007) A particle filter for joint detection tracking of color objects. Image Vis Comput 25(8):1271–1281
6. Doucet A, Vo BN, Andrieu C, Davy M (2002) Particle filtering for multi-target tracking and sensor management. Proc IEEE Int Conf Inform Fusion 1:474–481
7. Guan YP (2010) Spatio-temporal motion-based foreground segmentation and shadow suppression. IET Comput Vis 4(1):50–60
8. Huang Y, Long Y (2006) Super-resolution using neural networks based on the optimal recovery theory. Proceedings of IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing, 465–470
9. Long Y, Huang Y (2006) Image based source camera identification using demosaicking. Proceedings of IEEE Workshop on Multimedia Signal Processing, 419–424
10. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. Image Vis Comput 21(1):99–110
11. Ryoo MS, Aggarwal JK (2010) UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html
12. The EC Funded CAVIAR project/IST 2001 37540. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
13. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. Proc IEEE Comput Soc Conf Vis Pattern Recognit 1:511–518
14. Wang C, Tian M (2010) Passenger flow direction detection for public transportation based on video. Proceedings of IEEE International Conference on Multimedia Communications, 198–201
15. Wu B, Nevatia R (2007) Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. Int J Comput Vis 75(2):247–266
16. Xie D, Dang L, Tong R (2012) Video based head detection and tracking surveillance system. Proceedings of IEEE International Conference on Fuzzy System and Knowledge Discovery, 2832–2836

17. Yang M, Lv F, Xu W, Gong YH (2009) Detection driven adaptive multi-cue integration for multiple human tracking. Proceedings of IEEE International Conference on Computer Vision, 1554–1561
18. Yang T, Pan Q, Li J, Cheng Y, Zhao C (2004) Real-time head tracking system with an active camera. Proc IEEE World Congr Intell Control and Autom 3:1910–1914
19. Yuk JSC, Wong KK, Chung RH, Chin FY, Chow KP (2006) Real-time multiple head shape detection and tracking system with decentralized trackers. Proc IEEE Int Conf on Intell Syst Des Appl 2:384–389
20. Zhao M, Sun DH, Tang Y, He HP (2012) Head detection based on 21HT and circle existence model. Proceeding of the 10th World Congress on Intelligent Control and Automation, 4875–4880

R.-Y. Xu was born in Wenzhou, Zhejiang Province, China. He is now a M.D. Candidate of School of Communication and Information Engineering in Shanghai University, China. His major research interests include image processing, pattern recognition.

Y.-P. Guan was born in Xiaogan, Hubei Province, China, in 1967. He received the B.S. and M.S. degrees in physical geography from the Central South University, Changsha, China, in 1990, 2006, respectively, and the Ph.D. degree in geodetection and information technology from the Central South University, Changsha, China, in 2000.

From 2001 to 2002, he did his first postdoctoral research at Southeast University in electronic science and technology. From 2003 to 2004, he did his second postdoctoral research at Zhejiang University in communication engineering, and he had been an Assistant Professor with the Department of Information and Electronics Engineering, Zhejiang University. Since 2007, he has been a Professor with School of Communication and Information Engineering, Shanghai University. He is the author of more than 120 articles, and more than 20 patents. His research interests include intelligent information perception, digital image processing, computer vision, and security surveillance and guard.