PAPER

A Real-Time Human Detection System for Video

Bobo ZENG[†,††], ***Student Member***, **Guijin WANG**[†a)], ***Member***, **Xinggang LIN**[†], ***and*** **Chunxiao LIU**[†], ***Nonmembers***

SUMMARY This work presents a real-time human detection system for VGA (Video Graphics Array, 640×480) video, which well suits visual surveillance applications. To achieve high running speed and accuracy, firstly we design multiple fast scalar feature types on the gradient channels, and experimentally identify that NOGCF (Normalized Oriented Gradient Channel Feature) has better performance with Gentle AdaBoost in cascaded classifiers. A confidence measure for cascaded classifiers is developed and utilized in the subsequent tracking stage. Secondly, we propose to use speedup techniques including a detector pyramid for multi-scale detection and channel compression for integral channel calculation respectively. Thirdly, by integrating the detector's discrete detected humans and continuous detection confidence map, we employ a two-layer tracking by detection algorithm for further speedup and accuracy improvement. Compared with other methods, experiments show the system is significantly faster with 20 fps running speed in VGA video and has better accuracy as well.

key words: *human detection, human tracking, real-time detection, normalized oriented gradient channel feature*

1. Introduction

Automatically finding humans in images and videos has great importance for many applications such as the visual surveillance or advanced driver assistance systems, but it's challenging due to the large within-class variations caused by varying poses, illuminations, clothes and so on.

Many methods have been proposed for human detection, and the sliding window approach is the most effective as it exhaustively scans and discriminatively classifies the windows over positions and scales. It has two key components including feature and learning algorithm. Feature describes humans and learning algorithm discriminates humans and non-humans. The proposed human features can be categorized as shape features (HOG [1], Edgelet [2], etc.), texture features (LBP [3], etc.), color features (color similarity [4], etc.) and motion features (HOF [4], etc.). Among the above features, HOG feature is the most influential due to its strong discriminative power and moderate computational cost, and it has also been successfully applied to detect other objects apart from humans. A recent benchmark [5] reveals HOG still remains competitive, even though new features have been introduced subsequently. As for the learning algorithms, the most common ones are linear SVM and boosting. Detection accuracy can be improved by utilizing more complex features (such as covariance feature [6]), integrating heterogeneous features [4] or employing sophisticated learning algorithms (such as the intersection kernel SVM [7] or latent SVM [8]), but they suffer from greatly increased computation burden and are infeasible for real-time applications.

Some studies have focused on improving the detection speed. Generally, the sliding window methods are computationally intensive due to feature computation in multiple scales and classifier evaluation over huge numbers of windows. Some methods use the simple and fast Haar feature to realize early rejection, but this feature is known to be weak in discriminating humans and the performance may be damaged [9]. Instead of exhaustive scanning, a fast coarse to fine scanning method is proposed in Ref. [10] but it may fail to detect small humans. Some methods seek to speed up HOG. By eliminating the Gaussian mask and trilinear interpolation, a simplified but equally effective fast HOG feature is defined in Ref. [11], [12]. With the integral histogram technique [13], the variable-sized fast HOG features can be computed efficiently. Instead of the histogram feature, scalar channel feature is used to further speed up [14], [15] due to its simplicity. For classifier evaluation, cascaded classifiers structure [16] is the most effective, which rejects the majority of negative windows in early stages at low cost. However, despite all of the efforts devoted to improving the speed, the fastest method can run only 2.67 fps on 640×480 images for humans $\geq$ 50 pixels among the evaluated methods [17].

In video, human tracking is frequently utilized to find humans. With the advance in human detection, the tracking by detection method emerges as an effective tracking approach by employing the detector's output as an observation model. Okuma et al. [18] employ a detector for hockey players and track them in a particle filter tracking framework. Breitenstein et al. [19] utilize an off-line trained pedestrian detector [1] and online trained, instance-specific classifier via online boosting [20] for multi-person tracking-by-detection. Another frequently used technique in video is background modeling, and most real-time systems [21], [22] depend on it for a fast speed. For QVGA videos, the systems [21], [22] have achieved real-time performance. But for VGA videos, the background modeling itself costs much time, making real-time processing challenging. More importantly, background modeling applies to static cameras

Manuscript received November 25, 2011.
Manuscript revised March 15, 2012.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
[††]The author is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China.
a) E-mail: wangguijin@tsinghua.edu.cn

DOI: 10.1587/transinf.E95.D.1979

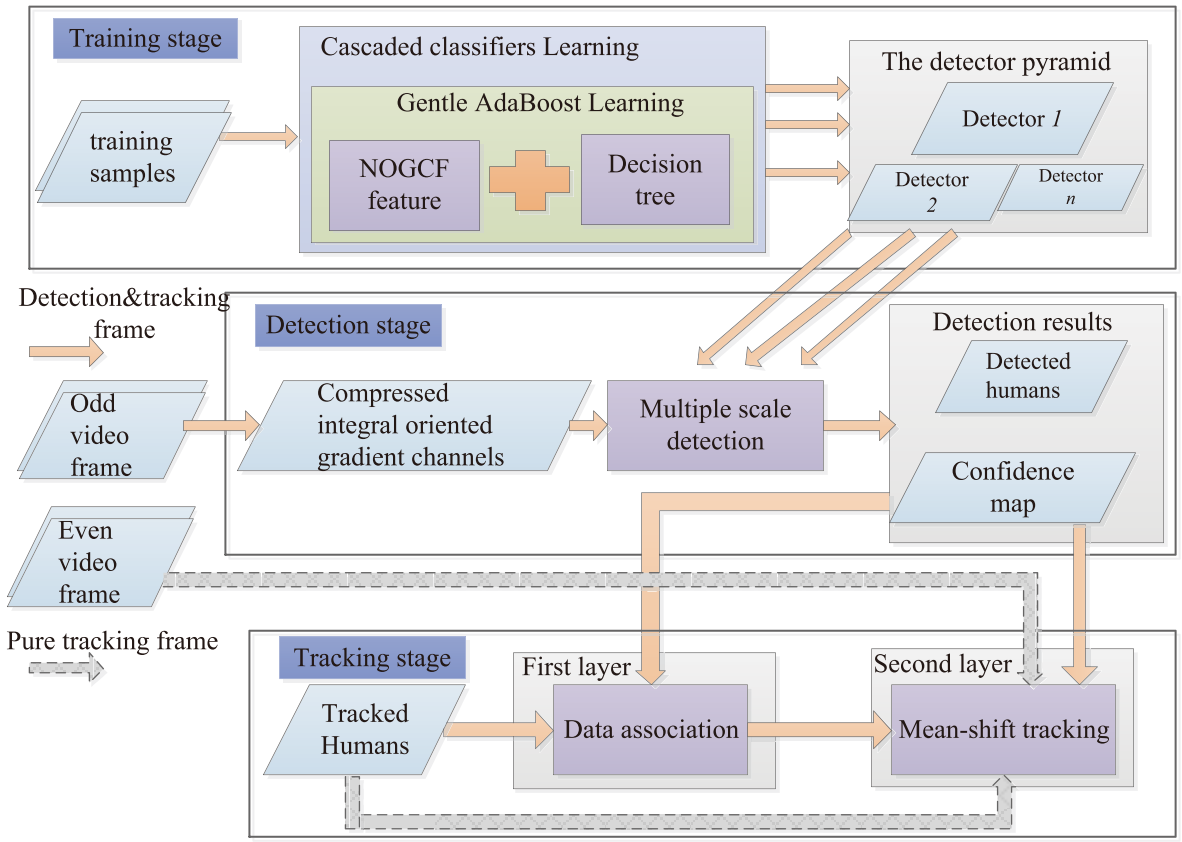

Fig. 1 Framework of the proposed system including training, detection and tracking stages.

only.

This work presents a human detection and tracking system which is mainly designed for some outdoor surveillance scenarios such as the unmanned toll gate of highway, where not many humans appear and the occlusion is not a problem. The most important achievement of the system is the real-time processing for VGA video on a common PC, without employing GPU or assuming static background, and it has high detection accuracy as well. The proposed system has three-fold contributions: (1) Inspired by Ref. [14], we extend their channel feature by constructing more scalar feature types on the oriented gradient channels and then experimentally identify the best feature type using the Gentle AdaBoost learning algorithm [23]. The proposed discriminative feature is fast and is the reason for the good performance of our system. (2) We integrate several fast detection techniques including a detector pyramid, the channel compression and gradient look-up table, which further speed up the detection. (3) We propose a confidence measure for evaluating the classification confidence of the cascaded classifiers, which enables the detector to output both the discrete detections and continuous confidence map. With the detector output's guidance, a fast tracking by detection method is introduced by combining object association and mean-shift tracking [24]. The final system runs 20 fps for the VGA resolution video, with a better accuracy than some state-of-the-art methods [1], [8], [12], as well as more than 10 times speedup.

The rest of this paper is organized as follows. In Sect. 2, the framework of the system including training, detection, and tracking is presented in overall. Section 3 explains the system's main modules in detail. Section 4 contains the experiments of our system. Section 5 gives a conclusion and the possible future work.

2. System Framework

The system has training, detection and tracking stages as illustrated in Fig. 1. The training stage learns the human detector, so it's the most important in determining the detection's speed and accuracy. We propose NOGCF (Normalized Oriented Gradient Channel Feature) as the human feature. It's a fast and discriminative scalar feature. A big NOGCF feature pool is generated by densely sliding on the human window in multiple positions and scales, for providing plenty of candidate features for learning. The Gentle AdaBoost learning algorithm, which is superior to Discrete AdaBoost and Real AdaBoost [23], is utilized to select the features. The weak classifiers, which are decision trees with the selected features, are combined into a strong classifier. Then similar to the VJ object detection framework [16], we train a cascade of strong classifiers as the human detector, which rejects the majority of negative windows in the early stages at low cost for fast classification. A detector pyramid

with different sized detectors is trained finally for the next detection stage.

In the detection stage, we use the trained detector pyramid to classify the sliding windows in the full image. The main focus of detection is the speed. Since the camera is not restrained to be stationary and may have pan/tilt/zoom motion, the foreground extraction for speedup by generating ROI is not used. For an input video frame, the oriented gradient channels are extracted, compressed and the corresponding integral channels are calculated for fast computation of features. Then the trained detector pyramid is performed on the channels to detect humans in multiple scales. The results are fused by a simple and fast nonmaximum suppression method [25]. It groups and merges detections which are adjacent both in position and scale. We also propose a confidence measure for the cascaded classifiers. Thus during the sliding window scanning, the continuous confidence map in multiple positions and scales is calculated. The discrete detections and continuous confidence map are fed into the next tracking stage.

The tracking stage doesn't aim to discriminate the humans' identity ; instead its main goal is improving the detection's speed and accuracy in three-fold. Firstly, since tracking is much faster than detection, we detect humans in odd frames only and use pure tracking to find humans in even frames, achieving a speedup of almost twice. Secondly, the occasionally appeared false positives in detection with no consistent temporal continuity can be eliminated. Thirdly, the missed detections between frames can be found back with temporal continuity and appearance similarity. For odd frames, to make the tracking fast, we propose a two-layer tracking by detection method. The first layer performs data association using motion prediction and appearance similarity, which almost costs no time. If a tracked human is successfully associated to a detected human, the tracked human is updated to the position of the detected result, so the tracking precision is guided by the detection result. In the case of missed detections of the detector, the first layer fails with no appropriate detection associated, then the second layer color histogram based mean-shift [24] tracking is performed. We choose mean-shift tracking method due to its fast tracking performance. For even frames, since no detection results are available, we use the mean-shift tracking directly.

3. System Modules

This section explains four important modules in the system in detail. Fast scalar feature types are designed and then a huge feature pool is generated. From the feature pool, the Gentle AdaBoost based learning algorithm selects the most discriminative features and the learned classifiers are organized in a cascaded form, with a corresponding confidence measure being formulated. The fast features and the learned classifiers are the foundation of the system's speed. Besides, additional speedup techniques are employed in the full image detection stage. Finally, the two-layer fast tracking by detection algorithm is utilized to improve the performance.

3.1 Scalar Feature Construction

HOG [1] or fast HOG [11] is a multi-dimensional histogram (e.g. 36 bins) and then inner product should be performed with SVM or LDA classifier, so it's computational intensive. Actually, many bins are not informative as the gradient orientations are sparse, and adding them may damage the performance of the classifier due to curse of dimensionality. Therefore, we propose to design scalar features on the oriented gradient channels to reduce the computational cost as well as maintaining the HOG feature's discriminative power.

Given the input image $I(x, y)$, its gradient magnitude $G(x, y)$ and orientation $O(x, y)$ is computed as

$$\begin{cases} G(x,y) = \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \\ O(x,y) = \arctan(I_y(x,y)/I_x(x,y)) + \frac{\pi}{2} \end{cases} \tag{1}$$

where $I_x(x, y), I_y(x, y)$ are the x, y derivative obtained with $[-1\ 0\ 1]$ mask. The gradient orientation is quantized into N_b bins $\left[(n-1)\frac{\pi}{N_b}, n\frac{\pi}{N_b}\right)$ with bin center $O_c(n)$, where $n = 1, \ldots, N_b$. $G(x, y)$ is divided into $O(x, y)$'s two circularly adjacent bins centered at $O_c(n)$ and $O_c(\mathrm{mod}(n, N_b) + 1)$ with linear interpolation as

$$\begin{cases} G_n(x,y) = G(x,y)\left(1 - \frac{O(x,y) - O_c(n)}{\pi/N_b}\right) \\ G_{\ \mathrm{mod}\ (n,N_b)+1}(x,y) = G(x,y)\frac{O(x,y) - O_c(n)}{\pi/N_b} \end{cases} \tag{2}$$

where $G_n(x, y)(1 \le n \le N_b)$ is oriented gradient channels. With the interpolation, aliasing is reduced and performance can be improved as stated in Ref. [1]. Also, an additional gradient magnitude channel $G_0(x, y) = G(x, y)$ is added for normalization. We define the feature $M_n(R)$ inside a rectangular region R on the channel n as

$$M_n(R) = \sum_{(u,v)\in R} G_n(u, v) \tag{3}$$

which is fast to compute with only 4 look-up and 3 addition/subtraction operations by means of integral image [16]. We call it Gradient Magnitude Channel Feature (GMCF) when $n = 0$ and Oriented Gradient Channel Feature (OGCF) when $n = 1, \ldots, N_b$. The two types of feature characterize the edge strength inside the specified rectangle (see Fig. 2 (a)).

HOG feature has a cell/block structure [1] for illumination normalization, where the gradient magnitude sum inside the cell rectangle R_c is divided by the sum inside the block rectangle R_b. Inspired by this, we design the Normalized Oriented Gradient Channel Feature (NOGCF) to be the 3rd feature type as follow:

$$\bar{M}_n(R_c, R_b) = \frac{M_n(R_c) + \varepsilon}{M_0(R_b) + \varepsilon}, n = 1, \ldots, N_b \tag{4}$$

where ε is a small value for avoiding 0 denominator. Though L_2 normalization has a slightly better performance [1], we select L_1 normalization as its computation cost is lower.

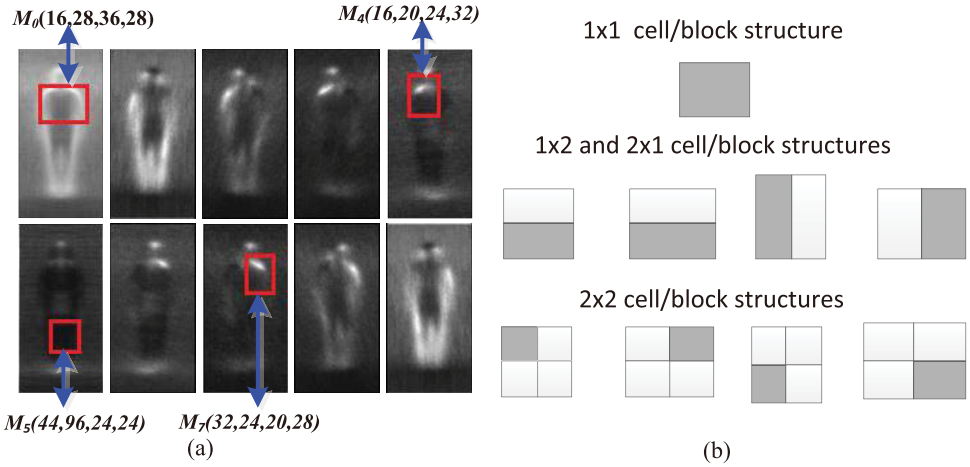

Fig. 2 Illustration of the proposed features: (a) Feature rectangles defined in oriented gradient channels for $N_b = 9$. The first channel is the gradient magnitude channel. The images are average gradient channels of human samples. (b) NOGCF features of different cell/block structures. The gray rectangle is the cell and the whole big rectangle is the block.

NOGCF characterizes the relative edge orientation strength inside the cell R_c to the gradient magnitude inside the block R_b. Cell and block have pre-defined geometric relation. In Ref. [1], a block is evenly divided into four cells on 2×2 grids. For enriching feature set, we add the 1×1, 2×1 and 1×2 grids as well (see Fig. 2 (b)).

GMCF, OGCF and NOGCF will be tested in the subsequent experiments to identify the best feature type or feature combination. A candidate feature pool is generated by sliding the feature rectangle in the human window. To reduce the feature pool size, the rectangle's location and size x, y, w, h are restricted to be divisible by 4. The generated feature pool is large with more than one million features, for providing sufficient number of potentially good features for the training.

3.2 Human Detector Learning

In the learning stage, a detector is trained for classifying a specified sized window $\mathbf{x}$ centered at (x, y). Gentle AdaBoost selects the most discriminative features from the feature pool to form a strong classifier. A cascade of strong classifiers $C(\mathbf{x})$ is trained as the human detector, with the strong classifier $C^l(\mathbf{x})$ in stage l $(0 \le l < L)$ has the form

$$C^l(\mathbf{x}) = \mathrm{sign}\left[H^l(\mathbf{x})\right] = \mathrm{sign}\left[\sum_{t=1}^{T} h_t^l(\mathbf{x}) - b^l\right] \tag{5}$$

where $h_t^l(\mathbf{x})$ is the weak regressor with real-valued outputs and b_l is the stage threshold for adjusting the detection rate and false positive rate of the strong classifier. Depth one regression tree is employed as the weak regressor $h_t^l(\mathbf{x})$ characterized by a triplet (a_l, a_r, θ) as

$$h_t^l(\mathbf{x}) = a_l \delta(f^{l,t}(\mathbf{x}) \le \theta) + a_r \delta(f^{l,t}(\mathbf{x}) > \theta) \tag{6}$$

where a_l, a_r are the regression values, θ is the split threshold, $f^{l,t}(\mathbf{x})$ is value of the selected feature in this weak regressor, and δ is the Kronecker delta function

In each stage of cascaded classifiers learning, weak regressor learned from the feature pool with the lowest regression error is added until the predefined minimum detection rate D_r and maximum false positive rate F_p are met. Since the feature pool is very large (over one million for $N_b = 9$) and exhaustively exploring all the features is infeasible, a small portion of features is randomly sampled with a rate $r = 0.02$ in each AdaBoost round. The whole training process is illustrated in Algorithm 1. We set D_r and F_p to be 0.999 and 0.5 respectively in the training.

Algorithm 1 Cascaded classifiers training.

Input:
Total cascade stages' number L;
Stage's minimum detection rate D_r;
Stage's maximum false positive rate F_p;
Feature random sampling rate r;
Sample number N;
Output:
Cascaded strong classifiers $C(\mathbf{x})$;
while stage number $l < L$ **do**
  $l = l + 1$;
  Select N positive and negative samples which are classified as positive with the already trained cascades.
  /*Train the strong classifier $C^l(\mathbf{x})$ using Gentle AdaBoost.*/
  while D_r and F_p are not satisfied **do**
    Randomly sample the candidate features from the entire feature pool according to r.
    Learn the weak regressor $h_t^l(\mathbf{x})$ by selecting the best feature and add it to $H^l(\mathbf{x})$.
    Update the sample weight.
  end while
end while

Besides the $-1/1$ discrete classified output of the cascaded classifiers, continuous confidence is also needed to judge the extent of a window belonging to the human, as the

confidence is beneficial for the subsequent tracking stage. For the SVM classifier used in Ref. [1], it's straightforward since the classifier has a continuous output and only one classifier is involved. But for the cascaded classifiers, many classifiers are cascaded one by one and it's inappropriate to use one particular classifier's output as the final confidence. We propose a confidence measure, under which the more stages the window passes, the higher confidence it will get, especially when it passes the last stage. Assume that the window passes stage l has the being human probability f_l, we can get a recursive formula based on D_r and F_p as

$$\begin{aligned} f_{l+1} &= \frac{f_l D_r}{f_l D_r + (1-f_l)F_p} = \frac{1}{1+\frac{1-f_l}{f_l}\frac{F_p}{D_r}} \\ &\approx \frac{1}{1+\frac{1-f_l}{f_l}F_p} \end{aligned} \tag{7}$$

The approximation is reasonable as D_r is close to 1. f_0 is the prior probability of being human and is set to 10^{-5}. For a window $\mathbf{x}$ with $C(\mathbf{x}) = -1$ which means it has passed some stage l and been rejected by stage $l+1$ ($l < L-1$), its confidence is measured by interpolation between f_l and f_{l+1} using $H^l(\mathbf{x})$, and then attenuated by a factor ω ($0 < \omega < 1$). Else for $C(\mathbf{x}) = 1$ which means it has passed all the stages, its confidence is enlarged by adding an additional term determined by $H^{L-1}(\mathbf{x})$. The measure is given as below

$$conf(\mathbf{x}) = \begin{cases} \left(f_l + \frac{1-e^{-H_l(\mathbf{x})/\rho}}{1+e^{-H_l(\mathbf{x})/\rho}}(f_{l+1} - f_l)\right)\omega \\ \quad \text{if } C(\mathbf{x}) = -1, C^l(\mathbf{x}) = 1, C^{l+1}(\mathbf{x}) = -1 \\ \omega f_{L-1} + \frac{(1-\omega)}{1+e^{-H_{L-1}(\mathbf{x})/\rho}} \quad \text{if } C(\mathbf{x}) = 1 \end{cases} \tag{8}$$

ρ is for tuning the sigmoid function. Given the confidence measure, all the sliding windows have a confidence after being classified, and a confidence map is generated.

3.3 Human Detection Speedup in the Full Image

In the previous two subsections, the fast feature and the cascaded classifiers are the foundation of the fast detection. In this subsection, we emphasize additional speedup techniques in the detection.

The Detector Pyramid for Fast Multi-scale Detection. Traditionally, only one detector is trained, so the multi-scale detection creates a densely sampled image pyramid, extracts features and performs classification in the images of all the scales (see Fig. 3 (a)). The creation of image pyramid and the feature extraction serves as a major bottleneck in the detection, especially when the feature calculation is computational intensive, such as the features in our system involving per-pixel gradient and orientation calculation. Dollár et al. [17] approximates the multiple nearby scales' gradient histograms given gradients computed at one scale. But the speedup is limited since the approximation is only effective inside one octave. Also it damages the performance a little. Besides, it cannot eliminate the time for computing the integral images of all the oriented gradient channels, which also accounts for much time.

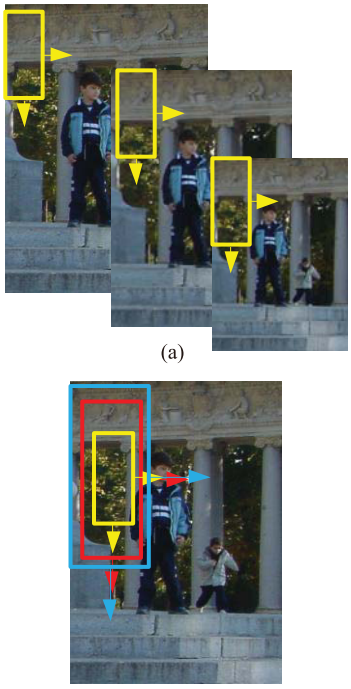

Fig. 3 Two kinds of multiple scale human detection: (a) Image pyramid. A detector with fixed size detects humans on multi-scaled image pyramid. (b) Proposed detector pyramid. A detector pyramid with multi-scaled sizes detects human on one image.

We propose to construct a detector pyramid by training different sized human detectors for multi-scale detection, so the pyramid detects different sized human in the same image (see Fig. 3 (b)). The oriented channels and integral images are computed only once and then shared among the detectors, therefore much time is saved. Actually, it's a method which sacrifices the training time for detection time. We train 5 human detectors of sizes $40 \times 80, 48 \times 96, 56 \times 112, 64 \times 128, 72 \times 144$. For QVGA resolution ($320 \times 240$), the pyramid can cover all the human sizes for most surveillance scenarios. For VGA resolution (640×480), we utilize a hybrid approach by running the detector pyramid on the VGA and resized QVGA resolution. In this case, human sizes range from 80 to 288 in height is covered without training new detectors.

Channel Compression. The oriented gradient channels of the detection image with size $M \times N$ can be compressed to $\frac{M}{4} \times \frac{N}{4}$ by summing the values in each 4×4 patch sequentially. The compression is reasonable since the feature rectangle (x, y, w, h) is 4-pixel aligned in the detection window. Also, the detection window slides in the detection image in 4-pixel step horizontally and vertically. Therefore all the features evaluated in the detection stage are 4-pixel aligned with the channels. Summing values inside (x, y, w, h) in the original channels is equivalent to summing inside $\left(\frac{x}{4}, \frac{y}{4}, \frac{w}{4}, \frac{h}{4}\right)$ in the compressed channels. For each $M \times N$ channel, $2MN$ additions are needed in computing the integral image originally. With this technique,

MN plus $\frac{1}{8}MN$ additions are needed for the compression and the compressed integral image computation, so the cost decreases by 40%. The technique has also been applied to the training in Sect. 3.2 to dramatically reduce the memory cost of the training samples to only $\frac{1}{16}$, which is essential for large sample training.

Lookup Table. We construct a lookup table for the gradient, orientation and linear interpolation in the calculation of $G_n(x, y)$ in Eq. (1)(2). The table's input is $I_x(x, y), I_y(x, y)$ where $(-255 \leq I_x(x, y), I_y(x, y) \leq 255)$ and the output is $(G_0, n_1, G_{n_1}, n_2, G_{n_2})$ including gradient G_0 and divided gradients G_{n_1}, G_{n_2} in the neighboring bins n_1 and n_2. By using lookup table, the time-consuming arctan and square root operations are eliminated and the speed can be improved.

3.4 Human Tracking

Pure detection is still hard to detect all the humans, so tracking with temporal information is utilized to retrieve missed detections and remove false positives. More importantly, in consideration of speed, detection is performed in the odd frames only and humans in the even frames are found with tracking at much low cost. The two-layer tracking by detection framework in the odd frames is illustrated in Fig. 4. The tracking is guided by both discrete detected humans and continuous human confidence map. The tracked humans get their new positions with motion prediction firstly, and then data association is carried out between the detected humans and tracked humans. If a tracked human is associated, its position is updated to the associated detection position. Else, it goes to the mean-shift tracking. The tracking has a confidence map guided termination judgement to decide whether to terminate the tracker. The un-associated detected humans go to the tracking initialization, where they are validated as true trackers or rejected as false positives. In the even frames without detection, only the mean-shift tracking module is performed.

Tracking Initialization. For a detection not associated to any existing trackers, a pre-tracker with the detection is initialized for a validation process. N frames are observed and the pre-tracked human should be detected for at least N_d frames. Also, the average detection confidence of the N_d detections should be above a threshold. The pre-tracker is turned into a tracker if it satisfies the requirements. By this confirmation, spurious detections can be rejected.

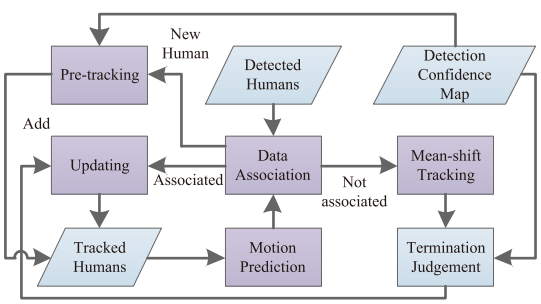

Fig. 4 Human tracking framework.

Data Association. Data Association links the detected results to the tracked results. To guarantee the association correctness, we combine human position, scale and color similarity measured as

$$S(T, D) = p_{\mathcal{N}}\left(\frac{pos_T - pos_D}{pos_T}\right) p_{\mathcal{N}}\left(\frac{size_T - size_D}{size_T}\right) B(H_T, H_D) \tag{9}$$

where T, D is the tracked humans and detected humans respectively. The first and second term measures the human position (center of human) and scale (human width and height) agreement between T and D, based on the observation the associated pairs should be similar both in position and scale. The similarity is measured by the Normal distribution $p_{\mathcal{N}}$. The third term measures the humans' appearance similarity by the widely used Bhattacharyya coefficient on color histograms [24].

A greedy association algorithm is employed to find the best matching pairs of tracking and detection. Firstly, a matching similarity matrix M of all the tracked and detected humans is calculated using Eqs. (9). Then the maximum similarity $S(T', D')$ in M is selected, and if it's greater than the matching threshold th, the pair is successfully associated and its row and column in M are deleted. Otherwise, no proper association exists. Repeat above association process until no further valid pair is available.

Two-layer Tracking. In the first layer, a constant velocity motion model is defined as

$$\begin{aligned} (x, y)_t' &= (x, y)_{t-1} + (u, v)_{t-1} \\ (u, v)_t &= (1 - \alpha)(u, v)_{t-1} + \alpha[(x, y)_t - (x, y)_{t-1}] \end{aligned} \tag{10}$$

where $(x, y)_t'$ is the predicted position and $(x, y)_t$ is the finally tracked position. The velocity update rate α is set to be 0.1. Data association is performed from position $(x, y)_t'$ to the detection results. If it's associated to a detection $D(x, y)$, $D(x, y)$ is taken as $(x, y)_t$. Otherwise, the second layer tracking is performed. Mean-shift iteration is started from $(x, y)_t'$ using RGB color histogram and the converged result is taken as $(x, y)_t$. Note that data association is in the first layer as it's very fast. If the speed is not a matter, we can also interchange the order by doing mean-shift tracking firstly and then doing data association.

Tracking Termination Guided by Confidence. It's very important to terminate a tracker in the right time. If it's terminated too early, the human is not tracked and the detection rate will drop. On the contrary, if terminated too late, a lot of false positives will be generated. We propose a detection confidence guided termination rule. The confidence T_c is looked up from confidence map (see Sect. 3.3) according to the tracker's position and scale. We set two thresholds th_{low} and th_{high}. If $T_c < th_{low}$, the tracker is immediately terminated (such as when the human exits). Else if $th_{low} \leq T_c \leq th_{high}$, the termination is determined by the tracker's number of consecutive frames of no association

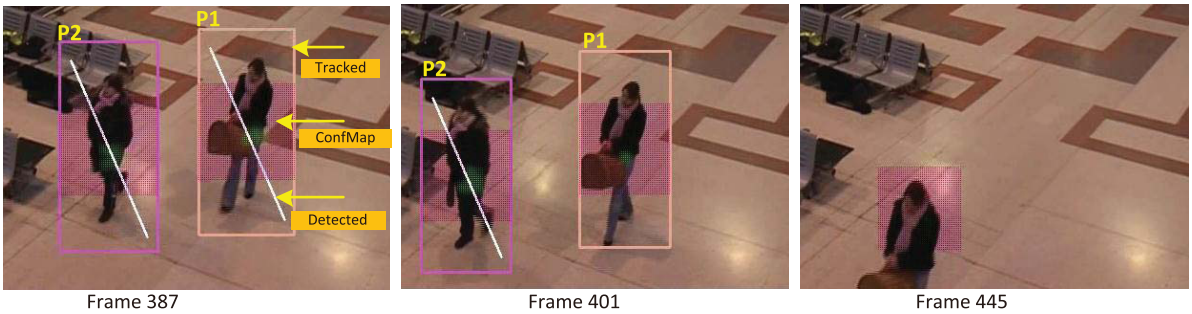

Fig. 5 The illustration of detection confidence guided termination rule in tracking. The rectangle and the diagonal line denote the tracked human and the detected human. Only the detection confidence in the around square is showed and green color means high value. **Left**: Human P1 has very high detection confidences around her and she is associated to a detection result. **Middle**: P1 is missed in the detection but the detection confidences are still high, so the tracker continues. **Right**: P1 leaves the scene and the detection confidences are very low, so the tracker is terminated.

N_{miss}. If $N_{miss} > N_{MAX}$, the tracker is terminated. Else, the tracking is continued. Else if $T_c > th_{high}$, we continue the tracker and set $N_{miss} = 0$. The detection confidence guided termination rule is very effective, since in most missed detection regions, the confidence is still higher than the background, and based on the rule the tracker can continue (see Fig. 5 for a detailed illustration). If judged by N_{miss} only, the tracker will soon be inappropriately terminated.

4. Experimental Results

4.1 Experiment Setup and Evaluation Criteria

We first evaluate the performance of proposed feature types (including NOGCF, GMCF and OGCF) to identify the best feature type or combinations for the detector, with the comparison to the HOG detector [1]. The evaluations are carried out on the INRIA [1] and Daimler [26] pedestrian dataset. Though Daimler dataset is captured on the running vehicle instead of in surveillance scenario, it can test proposed detector's performance for different applications. Then the whole system is evaluated on both our own surveillance video dataset and the public PETS 2006 dataset [27]. Our own dataset contains 10 VGA resolution videos with about 20 minutes long in total. For PETS 2006 dataset, we choose S4-T5-A-4 (2 minutes long) which has adequate number of humans and is similar to our application scenario. Apart from HOG, two additional state-of-the-art public available human detectors' results are given as a comparison (boosted Histogram [12] and cascaded Deformable [8]). The result of our previous fast detector (30 fps on QVGA) with matrix based structure [28] is also given. All experiments are carried out in a common PC with a dual-core 3.0 GHz processor.

Two evaluation criteria called FPPW and FPPI are employed in the literature [5]. FPPW (False Positive Per Window) evaluates a detector by classifying cropped human windows against densely generated negative windows from full images without human. As it doesn't consider the influence of post-processing or false detections on body parts, better per-window scores will not necessarily result in better per-image performance. Therefore, we use FPPI instead to evaluate the detector in the entire full images. In evaluation, a detection bb_{dt} with confidence c_{dt} is considered true if its overlap with the ground truth bb_{gt} satisfies the PASCAL criterion

$$\alpha_o = \frac{\text{area}(bb_{dt} \cap bb_{gt})}{\text{area}(bb_{dt} \cup bb_{gt})} > 0.5 \tag{11}$$

The miss rate vs. FPPI curve is obtained by continuously increase the confidence threshold of bb_{dt} at a small step.

4.2 Evaluation of the Detectors on Image Datasets

We train the human detectors on the INRIA dataset with NOGCF and its combination with other features. The trained detectors are evaluated on both INRIA and Daimler as illustrated in Fig. 6. Among the proposed features, NOGCF gains the best performance, while combining it with the other two feature types degrades the performance more or less. Though AdaBoost has the ability to select the most discriminative feature theoretically, the selection is based on the training set which may not well generalize to the testing set. Thus the fusing of multiple feature types using AdaBoost does not necessarily improve the performance. A similar phenomenon has also been observed in Ref. [29] when combining Haar feature. Due to enriched feature pool, NOGCF with all the structures improves over NOGCF with 2×2 cell/block structure only. From the results, NOGCF with all the structures is identified as our final feature type. Our detector is better than HOG on both datasets, especially with a large margin on Daimler.

To illustrate the speed of the detector, we list the number of NOGCF features and time cost units in the first 5 stages compared to the fast HOG [11] in Table 1, since both methods use cascaded classifiers structure whose speed is determined by several frontal stages. Each fast HOG feature is 36D and costs 36 time units per feature, while NOGCF costs only 1 time unit. 1 time unit is 8 table look-up, 8 addition and 1 division operations. The result shows NOGCF

maintains the discriminative power at a much low cost.

The speedup techniques in Sect. 3.3 are tested by using or removing them. The results are showed in Table 2. All techniques have improved the speed considerably, and the overall speedup is 55.9%, which is significant.

4.3 Evaluation of the System on Video Datasets

To further improve the performance, we train the detector pyramid by enlarging the positive samples in INRIA from 2416 to 5000. We train each detector to 10^{-5} in FPPW to achieve a low false positive rate. The detector's accuracy and speed evaluated on our video and PETS 2006 dataset is illustrated in Table 3, given the comparative results of HOG, boosted Histogram and cascaded Deformable. Occlusion is not considered in all the evaluations. The results show our method improves significantly in both accuracy and speed. Our detector with detector pyramid has the same accuracy with the image pyramid version, but the speed is twice faster, which illustrates the better performance of the detector pyramid. Humans in PETS 2006 are more in a top view rather than a frontal view in our dataset, so the detection rate drops. With the effective tracking method, detection rate increases on both datasets by 5% and 15% respectively. The system with detection and tracking run in real-time with 20 fps.

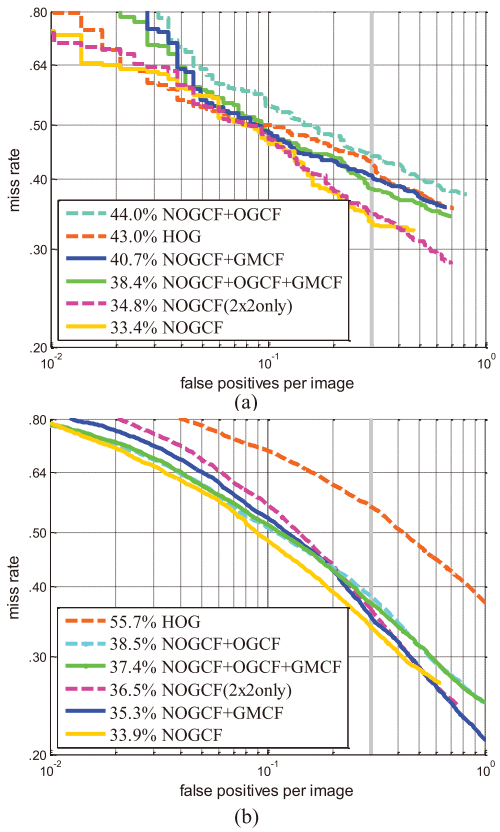

Fig. 6 Detectors's performance on (a) INRIA and (b) Daimler datasets with different feature configurations, given the comparison of HOG. Missed detection percentages at 0.3 FPPI is listed in the parenthesis.

We evaluate the tracking module on our dataset. The average tracking time for the odd frames with detection (means two-layer tracking) and even frames without detection (means mean-shift tracking) is 0.9 ms and 4.3 ms perspectively, so on only 2.6 ms is spent on the tracking. If we interchange the order of data association and mean-shift tracking, the accuracy (FP:98.31%,FPPI:2.89%) has no noticeable improvement, but the tracking time increases to 4.7 ms. This validates the ordering of the two-layer tracking.

Finally, Fig. 7 shows some real examples on INRIA dataset (detection only), our own video dataset and a sequence from PETS 2006 dataset.

Table 1 Number of features and time cost units for the features in first 5 stages of the trained cascaded classifier by NOGCF and fast HOG respectively. Each NOGCF and fast HOG needs 1 and 36 time units to compute respectively.

Method	Stage 0	Stage 1	Stage 2	Stage 3	Stage 4
#NOGCF,time	4,4	12,12	12,12	18,18	18,18
#Fast HOG,time	2,64	5,160	5,160	8,256	13,416

Table 2 Speedup of the techniques on 640×480 images. T_1: detector pyramid. T_2: channel compression. T_3: lookup table.

Techniques	None	With T_1	With T_2	With T_3	With all
Time (ms)	211	149	171	153	93
Speedup		29.4%	19.0%	27.5%	55.9%

Table 3 The detection rate (DR), false positive per image rate (FPPI) and running time per frame (Time) on our video dataset and PETS 2006. For our methods, the results of detection with the image pyramid, with the detector pyramid and the final system are given. Three state-of-the-art methods are evaluated for comparison.

Dataset	Our dataset			PETS 2006		
Method	DR	FPPI	Time	DR	FPPI	Time
cascaded LatSvm [8]	70.35%	9.20%	1679 ms	61.4%	7.3%	1810 ms
boosted Hist [12]	77.34%	8.30%	1058 ms	63.5%	8.6%	1098 ms
matrix Structure [28]	79.65%	6.25%	186 ms	70.3%	9.1%	195 ms
HOG [1]	84.84%	3.64%	1238 ms	75.1%	7.4%	1310 ms
Our detector (image pyramid)	93.13%	2.10%	210 ms	81.5%	7.3%	221 ms
Our detector (detector pyramid)	93.69%	2.30%	100 ms	81.3%	7.1%	105 ms
Our system with detection&tracking	**98.21%**	**2.76%**	**51 ms**	**96.0%**	**3.7%**	**53 ms**

Fig. 7 Some detection and tracking examples. **Row 1**: the pure detection examples on the INRIA dataset. **Row 2**: system examples on our dataset. **Row 3**: system examples with snow on our dataset. **Row 4**: system examples on PETS 2006 dataset.

5. Conclusion

In this paper, we present a real-time human detection system for video. Experimental results illustrate its high accuracy and a running speed of 20 fps in VGA video, better than the existing state-of-the-art methods. Its superiority is attributed to the proposed NOGCF feature, speedup techniques such as the detector pyramid and the fast tracking method. In future research, we plan to introduce more channels such as the color channels to improve the detection further. Besides, we will extend the system to driver assistant applications, which requires an advanced tracking method for handling fast camera motion.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No.61132007).

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.886–893, 2005.
[2] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," Int. J. Comput. Vis., vol.75, no.2, pp.247–266, 2007.
[3] X. Wang, T.X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," 2009 IEEE 12th International Conference on Computer Vision, pp.32–39, 2009.
[4] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1030–1037, 2010.
[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," CVPR, June 2009.
[6] S. Paisitkriangkcrai, C.H. Shen, and J. Zhang, "Fast pedestrian detection using a cascade of boosted covariance features," IEEE Trans. Circuits Syst. Video Technol., vol.18, no.8, pp.1140–1151, 2008.
[7] S. Maji and A.C. Berg, "Classification using intersection kernel support vector machines is efficient," 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[8] P.F. Felzenszwalb, R.B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2241–2248, 2010.
[9] C. Wojek and B. Schiele, "A performance evaluation of single and multi-feature people detection," DAGM-Symposium, pp.82–91, 2008.
[10] Z. Wei, G. Zelinsky, and D. Samaras, "Real-time accurate object detection using multiple resolutions," 2007 IEEE 11th International Conference on Computer Vision, ICCV, pp.1–8, 2007.
[11] Z. Qiang, Y. Mei-Chen, C. Kwang-Ting, and S. Avidan, "Fast hu-

man detection using a cascade of histograms of oriented gradients," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.1491–1498, 2006.
[12] I. Laptev, "Improving object detection with boosted histograms," Image Vis. Comput., vol.27, no.5, pp.535–544, 2009.
[13] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.829–836, 2005.
[14] P.P.P. Dollár, Z. Tu, and S. Belongie, "Integral channel features," BMVC, 2009.
[15] W. Gao, H.Z. Ai, and S.H. Lao, "Adaptive contour features in oriented granular space for human detection and segmentation," CVPR: 2009 IEEE Conference on Computer Vision and Pattern Recognition, vol.1-4, pp.1786–1793, 2009.
[16] P. Viola and M. Jones., "Rapid object detection using a boosted cascade of simple features," Coference on Computer Vision and Pattern Recognition (CVPR), pp.511–518, 2001.
[17] S.B.P. Dollár and P. Perona, "The fastest pedestrian detector in the west," BMVC, 2010.
[18] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," Computer Vision-ECCV 2004, pp.28–39, 2004.
[19] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L.V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," IEEE International Conference on Computer Vision, Oct. 2009.
[20] H. Grabner and H. Bischof, "On-line boosting and vision," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.260–267, 2006.
[21] Z. Jianpeng and H. Jack, "Real time robust human detection and tracking system," IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops, pp.149–149.
[22] A. Garcia-Martin and J.M. Martinez, "Robust real time moving people detection in surveillance scenarios," 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp.241–247, 2010.
[23] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," Annals of Statistics, vol.28, no.2, pp.337–374, 2000.
[24] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.5, pp.564–577, 2003.
[25] D.M. Pedro, F. Felzenszwalb, Ross B. Girshick, and D. Ramanan, "Object detection with discriminatively trained part based models," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.9, pp.1627–1645, 2010.
[26] M. Enzweiler and D.M. Gavrila, "Monocular pedestrian detection: Survey and experiments," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.12, pp.2179–2195, 2009.
[27] "http://www.cvg.rdg.ac.uk/pets2006/"
[28] G. Pang, G.J. Wang, and X.G. Lin, "Real-time human detection using hierarchical hog matrices," IEICE Trans. Inf. & Syst., vol.E93-D, no.3, pp.658–661, March 2010.
[29] S. Walk, K. Schindler, and B. Schiele, "Disparity statistics for pedestrian detection: Combining appearance, motion and stereo," Computer Vision - ECCV 2010, vol.6316, pp.182–195, 2010.

Bobo Zeng was born in 1985. He received his B.S. degree in the department of Electronics and information Engineering, Huazhong University of Science & Technology, Wuhan, China in 2007. He is currently pursuing the Ph.D. degree at Department of Electronics Engeering, Tsinghua University, China. His research interests include hand gesture detecion & recognition, pedestrian detection and tracking, surveillance video analysis, etc.

Guijin Wang was born in 1976. He received the B.S. and Ph.D. degree (with honor) from the department of Electronics Engineering, Tsinghua University, China in 1998, 2003 respectively, all in Signal and Information Processing. From 2003 to 2006, he has been with Sony Information Technologies Laboratories as a researcher. From Oct., 2006, he has been with the department of Electronics Engineering, Tsinghua University, China as an associate professor. He has published over 30 International journal and conference papers, hold several patents. He is the session chair of IEEE CCNC'06, the reviewers for many international journals and conferences. His research interests are focused on wireless multimedia, mesh network, image and video processing, object detection and tracking, online learning, etc.

Xinggang Lin received B.S. degree in Electronics Engineering, Tsinghua University, China in 1970; M.S. degree in 1986 and Ph.D. degrees in 1982, both in information science, Kyoto University, Japan. He joined the Department of Electronics Engineering at Tsinghua University in 1986 where he has been a full professor since 1990. He received "Great Contribution Award" from Ministry of Science and Technology of China, and "Promotion Awards of Science and Technology" from Beijing Municipality. He was a General Co-chair of the second IEEE Pacific-Rim Conference on Multimedia, an associate editor of IEEE T. on CSVT, and a technical/organizing committee member of many international conferences. He is a fellow of China Institute of Communications, and he published over 140 referred conference and journal papers in diversified research fields.

Chunxiao Liu is a Ph.D. student in the Department of Electronics Engineering, Tsinghua University, China. He received his B.S. degree in the department of Electronics and Information Engineering, Huazhong University of Science & Technology, Wuhan in 2008. His research interests include human re-identification, tracking, camera network, etc.