Object Tracking across Non-overlapping Cameras Using Adaptive Models

Xiaotang Chen, Kaiqi Huang, and Tieniu Tan

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
{xtchen,kqhuang,tnt}@nlpr.ia.ac.cn

Abstract. In this paper, we propose a novel approach to track multiple objects across non-overlapping cameras, which aims at giving each object a unique label during its appearance in the whole multi-camera system. We formulate the problem of the multiclass object recognition as a binary classification problem based on an AdaBoost classifier. As the illumination variance, viewpoint changes, and camera characteristic changes vary with camera pairs, appearance changes of objects across different camera pairs generally follow different patterns. Based on this fact, we use a categorical variable indicating the entry/exit cameras as a feature to deal with different patterns of appearance changes across cameras. For each labeled object, an adaptive model describing the intraclass similarity is computed and integrated into a sequence based matching framework, depending on which the final matching decisions are made. Multiple experiments are performed on different datasets. Experimental results demonstrate the effectiveness of the proposed method.

1 Introduction

In recent years, multi-camera visual surveillance systems have attracted much attention and been widely used in applications such as continuously tracking interested objects, early warning of abnormal events, etc. Using multiple cameras, the area of surveillance is expanded and the occlusion problem which is very challenging for single cameras can be solved to a certain degree. To continuously track objects in wide areas, the key problem is to establish correspondences of different object sequences observed under multiple cameras with either overlapping or non-overlapping views. To solve this problem, the overlapping multi-camera tracking systems generally require calibration of cameras and make use of geometric constrains (e.g. homography constrains) [1, 2]. While tracking objects across non-overlapping views is more challenging, for the observations under different cameras are widely separated in both time and space because of blind areas between cameras. In this paper, we focus on object tracking across multiple non-overlapping cameras.

Because few of spatio-temporal cues can be used in tracking multiple objects across non-overlapping cameras, the tracking decisions depend heavily on object matching. However, the appearance changes greatly across cameras due to

J.-I. Park and J. Kim (Eds.): ACCV 2012 Workshops, Part II, LNCS 7729, pp. 464–477, 2013.

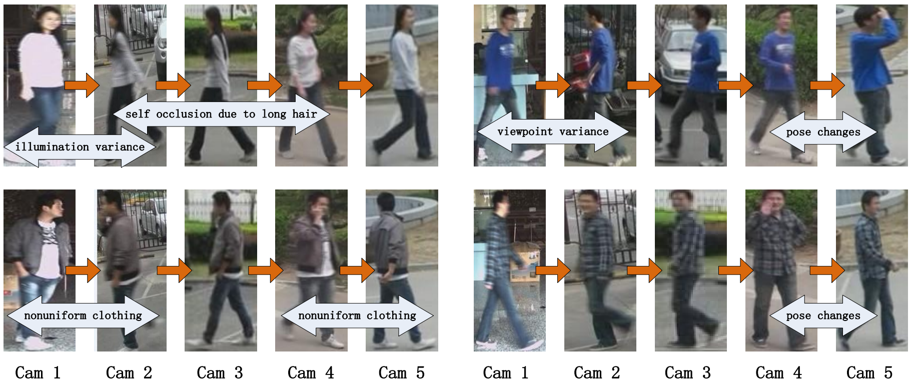

Fig. 1. Observations under multiple cameras with non-overlapping views. Each person walks along the same path from Cam 1 to Cam 5.

many factors, such as illumination variance, viewpoint changes, pose changes, nonuniform clothing, self-occlusions, different camera characteristics, as shown in Figure 1. The appearance changes of different persons across the same camera pair (e.g. from Cam 1 to Cam 2, or from Cam 2 to Cam 3) generally have the same pattern. For example, the appearances suffer from illumination variance and viewpoint changes between Cam 1 and Cam 2, while they almost remain the same between Cam 2 and Cam 3. In other words, the pattern of appearance changes varies according to the entry/exit cameras (the camera the object exits from and the camera it enters into). Based on this fact, various methods have been proposed over the last few years to deal with appearance changes across cameras. Methods [3–5] present brightness transfer functions or color transfer approaches to transform color between each pair of entry/exit cameras. In [6], camera transfer functions are applied to transform not only color but also other features. In this paper, we also deal with the appearance changes differently according to entry/exit cameras. Unlike previous work, we take entry/exit cameras as a categorical variable (or a feature) in the AdaBoost algorithm rather than transform the appearance from one camera to another.

Many methods [7–11] take the problem of object matching across cameras as object re-identification. Method [7] extracts color and texture histogram features to represent the objects, and uses machine learning tools to learn the similarity between any two objects. D'Angelo, A. *et al.* [8] propose a probabilistic color histogram based on a fuzzy K-Nearest Neighbors classifier. Xiaotang, C. *et al.* [9] present a direction-based stochastic matching method using directional cues of objects. These object re-identification methods always return a best match in a dataset for a given object, assuming the given object exists in the dataset. However, for a real multi-camera object tracking system, the number of moving objects is uncertain and an object with a new identity may appear at any time, in which case the identity of this new object can never be defined as anyone in the dataset, thus, defining each newcomer as the best match found by object re-identification methods can not always work in the real multi-camera object tracking systems. Obviously, directly setting a fixed threshold of the output similarities for all the objects is not proper. To solve this problem, the proposed method learns an adaptive model for each object to help drawing the final matching decisions.

Overall, the contributions in this paper lie in two aspects: (1) taking the entry/exit cameras as a feature to deal with the problem of appearance changes varying with entry/exit cameras; (2) using an adaptive model for each object in tracking objects across cameras instead of setting common rules for all the objects. Note that AdaBoost is used as a black-box machine learning tool. Other machine learning tools can also be used coherently with our framework.

Figure 2 shows the flowchart of the proposed system. Firstly, a blob sequence is initialized for an object when it comes into view, then after the initialization is completed, the blob sequence is matched against other objects which have left the views in a sequence based matching framework in order to determine whether this newcomer has been previously tracked or not. Once its identity is specified, it is continuously tracked under the single camera until it leaves. Finally, the dataset is updated by this labeled object.

This paper is organized as follows. In Section 2, the strategy of single camera object tracking used in our multi-camera system is presented. Section 3 describes a novel approach of tracking objects across non-overlapping cameras. Experimental results and conclusions are given in Section 4 and Section 5 respectively.

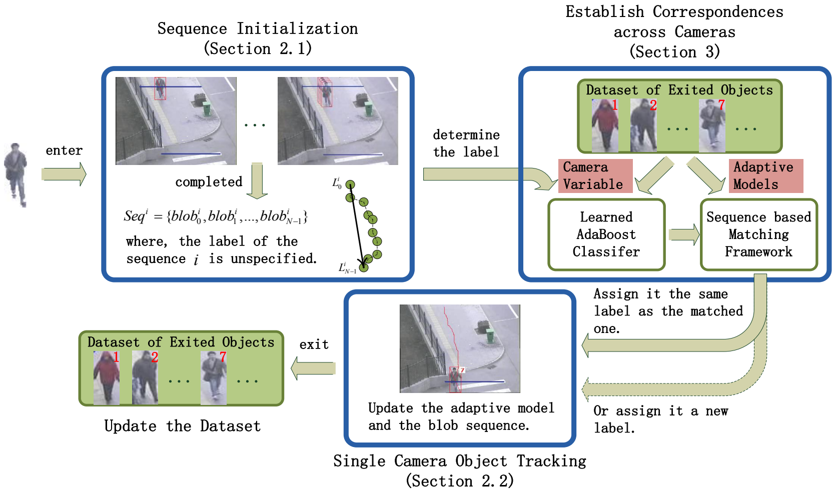

Fig. 2. The flowchart of our method

2 Single Camera Object Tracking

The task of single camera object tracking is to establish correspondences of blobs in successive frames under single cameras.

2.1 Sequence Initialization

In this paper, we use entry/exit lines to mark the locations of entry/exit zones. Each object entering the field of view of a camera must cross an entry/exit line, which initializes a sequence of blobs, as shown in Figure 2. If the diagonal line (from upper left to lower right) of the bounding box of a blob intersects an entry/exit line, then we believe this blob, denoted by $blob_n$, is detected in the entry/exit zones. The correspondences between $blob_n$ and uninitialized blob sequences $\{unSeq^i | unSeq^i = \{blob_0^i, blob_1^i, ...blob_{n-1}^i\}, n < N\}$ are established by finding the nearest location:

$$i^* = \arg\min_i \left\{ Dis(L_n, L_{n-1}^i) | Dis(L_n, L_{n-1}^i) < D_T \right\} \tag{1}$$

where i is the unspecified label of the blob sequence and D_T is a given threshold. L_n and L_{n-1}^i denote the locations of $blob_n$ and the latest blob in $unSeq^i$. When a match is found, $blob_n$ is used to extend $unSeq^{i^*}$. Otherwise, a new $unSeq^i$ is established. As long as the sequence length is up to N, the initialization of this sequence is completed. To be clear, $unSeq^i$ refers to the blob sequence which has not finished initialization yet. Once its initialization is completed, a label (or identity) is determined and assigned to it by establishing correspondences across cameras (described in Section 3). Then $unSeq^i$ becomes Seq^i, which is an initialized blob sequence with a specified label.

2.2 Establishing Correspondences under Single Cameras

For a blob detected in non-entry/exit zones, denoted by $blob_N$, we match it against initialized blob sequences $\{Seq^i\}$, where the label i has been specified, under the same camera using only directional cues. In Figure 3, each blob sequence has a direction vector, from the earliest location L_0 to the latest location L_{N-1}, roughly giving the moving direction of the corresponding object. Under a reasonable assumption that the moving direction of the same object does not change greatly in successive frames, $blob_N$ is assigned to Seq^{i^*} as:

$$i^* = \arg\min_i \left\{ \theta_i | \theta_i = \angle \left(\overrightarrow{L_0^i L_{N-1}^i}, \overrightarrow{L_0^i L_N} \right), \theta_i < \theta_T \right\} \tag{2}$$

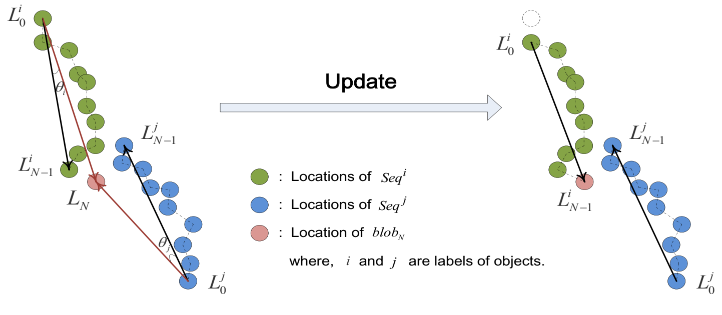

Fig. 3. Establishing correspondences under single cameras

where L_N is the location of $blob_N$. θ_T is a given threshold. Then, $blob_N$ is used to update Seq^{i^*} and its direction vector, as shown in Figure 3. The blobs detected in entry/exit zones are dealt with in the same way as the blobs in non-entry/exit zones first. Then it is matched against $\{unSeq^i\}$ as mentioned in Section 2.1, if the first round of match fails.

Establishing correspondences under single cameras is done depending only on the directional cues, not the appearance cues, making it efficient and fast when applied to tracking multiple objects in a large-scale multi-camera system.

3 Establishing Correspondences across Cameras

Once a blob sequence completes initialization, we match it against labeled objects having left the connected entry/exit zones to define the identification (or label) of this blob sequence.

3.1 Learning the Adaptive Model

For each blob, we only deal with the area occupied by the object. Several appearance-based features are explored to represent the blob. They are described in the following:

MCSH. In RGB color space, a major color spectrum histogram (MCSH) is computed for the blob, and the similarity is measured using [12]. The number of major colors is 50. The threshold of color distance is set to 0.06.
Histogram. The color histogram (H) of the blob is used, and normalized by the size of the object in pixels. Distance is measured using histogram intersection.
Major Colors. To incorporate color spatial information into the representation, the blob is partitioned into three parts using the method in [13], corresponding to the head, the upper body, and the lower body. A major color is clustered using K-means for the upper body (MU) and lower body (ML) respectively. Euclidean Distance is used to measure the distance between two colors.
Spatiogram. Spatiogram (S) [14] is also applied for the purpose of describing the spatial distribution of color. The similarity between two spatiograms is computed as the weighted sum of the similarity between two histograms.
LBP. LBP [15] is used as a texture feature, and comparison is done by the Bhattacharyya distance.

As mentioned above, the similarity vector between $blob^i$ and $blob^j$ is measured as $\{Sim^{ij}_{MCSH}, Sim^{ij}_H, Sim^{ij}_{MU}, Sim^{ij}_{ML}, Sim^{ij}_S, Sim^{ij}_{LBP}\}$, or Sim^{ij} for short, where i and j are labels of the corresponding objects. To deal with different patterns of appearance changes across cameras, we use a categorical variable E^iE^j in addition, indicating the camera E^i whose view $blob^i$ enters into and the camera E^j which $blob^j$ exits from. In the training process, the vector $[Sim^{ij}, E^iE^j]$ computed from two blobs corresponding to the same object (namely, $i = j$) is taken as a positive sample, while the negative sample set corresponds to different objects. Thus, the multiclass object recognition problem is formulated as a binary classification problem. The AdaBoost classifier is used to solve this problem with decision trees as weak classifiers. The output of AdaBoost is a weighted sum over the learned weak classifiers. The larger the value the more similar the two blobs. Different with other AdaBoost algorithms, we use learned adaptive thresholds for different objects to deal with the outputs instead of using a sign function.

An adaptive model for each labeled object is learned by calculating and updating the adaptive threshold. It is calculated based on the similarities of two blobs in the initialized blob sequence and updated over time, thus it indicates the intraclass similarity of the corresponding object to a certain degree. Given the initialized blob sequence $\{blob_0^i, blob_1^i, \cdots, blob_{N-1}^i\}$ of $Object^i$ (namely Seq^i), the adaptive threshold is computed as:

$$Thre^i = \frac{1}{N-1} \sum_{n=0}^{N-2} f\left(\left[Sim(blob_n^i, blob_{n+1}^i), E^i E^i\right]\right) \tag{3}$$

where $f(*)$ is the learned AdaBoost classifier.

When $Object^i$ is continuously tracked under Camera E^i, we update Seq^i, as described in Sec.2.2, as well as $Thre^i$:

$$Thre^i \leftarrow \frac{f\left(\left[Sim(blob_{N-1}^i, blob_N), E^i E^i\right]\right) + Thre^i}{2} \tag{4}$$

where $blob_N$ is the blob which updates Seq^i.

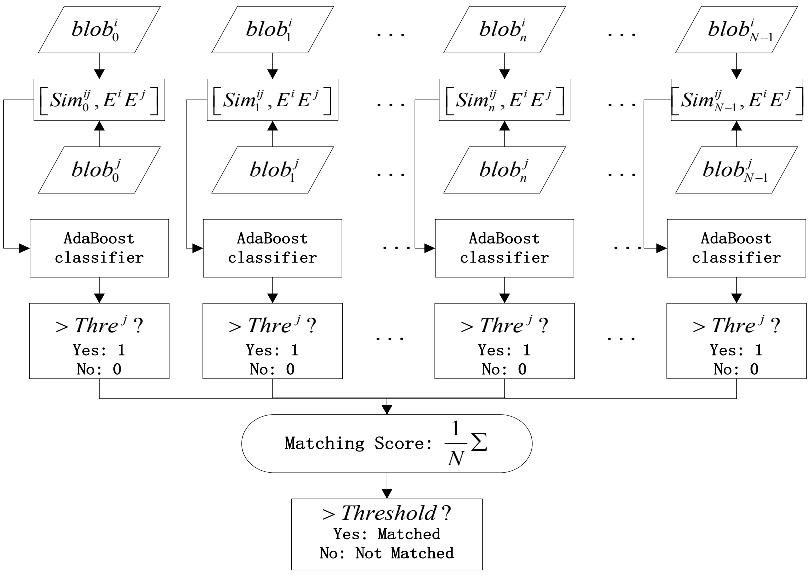

Fig. 4. Sequence based matching framework

3.2 Object Matching across Cameras

Following the work in [16], the weighted cross correlated model is used to estimate the topology of a multi-camera network. And MCSH [12] is applied to this model, by which connectivity and average transition time between two entry/exit zones are learned.

Given Seq^i with an unspecified label i, which completes initialization at time t, we match it against labeled objects (i.e. $Object^j$) leaving the connected entry/exit zones during the time window $[t - T_{Z^iZ^j} - T,\ t - T_{Z^iZ^j} + T]$, where Z^i and Z^j denotes the entry/exit zone where Seq^i detected and the entry/exit zone which $Object^j$ exits from respectively. T allows a small fluctuation around the average transition time $T_{Z^iZ^j}$.

Figure 4 shows the sequence based matching framework. Firstly, a feature vector is extracted from each blob. Secondly, a similarity vector is calculated and the camera variable E^iE^j is obtained as well. Finally, a decision is made according to the outputs of AdaBoost, where $Thre^j$ is the adaptive threshold of $Object^j$. If no one successfully matches Seq^i, then $Object^i$ is identified to be a new object and a new label is assigned to it. Otherwise, it retains the same label as the one with a maximal matching score. This framework is based on the assumption that if $blob_n^i$ and $blob_n^j$ belong to the same object, the similarity between them agrees with the similarity between two blobs of $Object^i$ or $Object^j$, as the appearance changes across cameras have been considered by using E^iE^j. $Thre^j$ has been updated over time along with Seq^j, so it is relatively stable and represents the similarity between every two blobs of $Object^j$. $Thre^i$ of Seq^i can also be calculated using Eq.3, however, it is relatively unstable as it has not been updated over time and could be greatly influenced by noise (false detections). Thus, we choose $Thre^j$ for comparison rather than $Thre^i$.

4 Experimental Results and Analysis

The first experiment is conducted to show the performance of the proposed object matching method using the AdaBoost classifier based on only color and texture features, in which samples are not representations of single blobs, but similarity measures between them. To compare with other methods, the experiment is based on the VIPeR dataset [17].

The dataset contains 1264 images of 632 pedestrians and each pedestrian has two images seen from different views. Since no camera information is given by this dataset, we train the AdaBoost classifier without using the categorical variable E^iE^j. As done in [7], we randomly split the dataset into two halves: the training set and the testing set. We show the average of the results on multiple random train-test sets and compare it to 5 different benchmark methods[1], Template (sum-of-squared distances matching), Histogram, Hand Localized Histogram [18], Principal Axis Histogram [2], and ELF 200 [7]. The results are presented using cumulative matching characteristic (CMC) curves, as shown in

[1] Results of other methods are from the work [7].

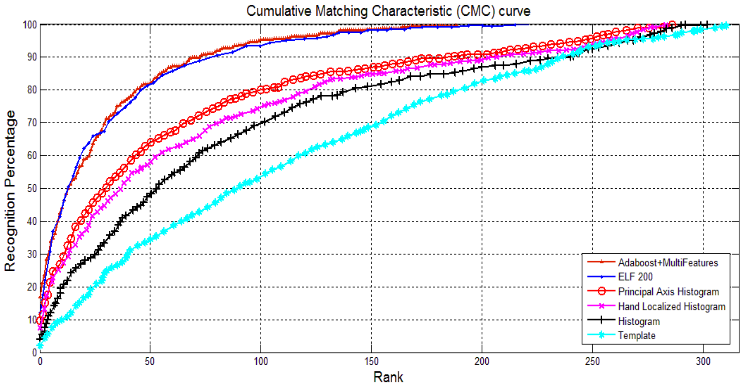

Fig. 5. CMC curves of different methods

Figure 5. It indicates that the performance of our method matches or exceeds other methods, and the rank 1 matching rate is about 16.77%. Figure 6 shows some examples of the matching results using our method. However, this experiment can not fully reveal the performance of our method, for the camera variable $E^i E^j$ is not used which provides important information and can greatly improve the performance of object recognition across cameras.

To the best of our knowledge, there is no public dataset providing images of pedestrians observed under multiple cameras (no less than three cameras). Thus, to demonstrate the effectiveness of the camera variable $E^i E^j$ for automated recognition, we conduct the second experiment based on a dataset[2] which consists of manually labeled pedestrians from off-line videos. This dataset contains 585 images of 39 pedestrians seen from three cameras with non-overlapping views, among which 25 pedestrians walk across cameras. Each pedestrian has 9 images under a single view. Some examples of the dataset is shown in Figure 9. As the samples are similarity measures between any two images from the same or different cameras, we collect 2358 positive samples and 13203 negative samples for the training set, and 1251 positive samples and 10530 negative samples for the testing set. The results are presented using receiver operator characteristic (ROC) curves in Figure 7. Our method using the camera variable $E^i E^j$ outperforms the one that does not using it, demonstrating the effectiveness of the camera variable.

To demonstrate the performance of our method on object tracking across cameras, we conduct the third experiment based on a non-overlapping multi-camera system both indoors and outdoors. Figure 8 (a) shows the layout of the system. The learned connections and average transition time between entry/exit zones across cameras are shown in Figure 8 (b).

The dataset used to train the AdaBoost classifier is the same dataset in the second experiment. The proposed object tracking algorithm is tested on videos[3]

[2] This dataset is available on the website: `http://www.datatang.com/Member/76804`

[3] The videos are available on the website: `http://www.datatang.com/Member/76804`

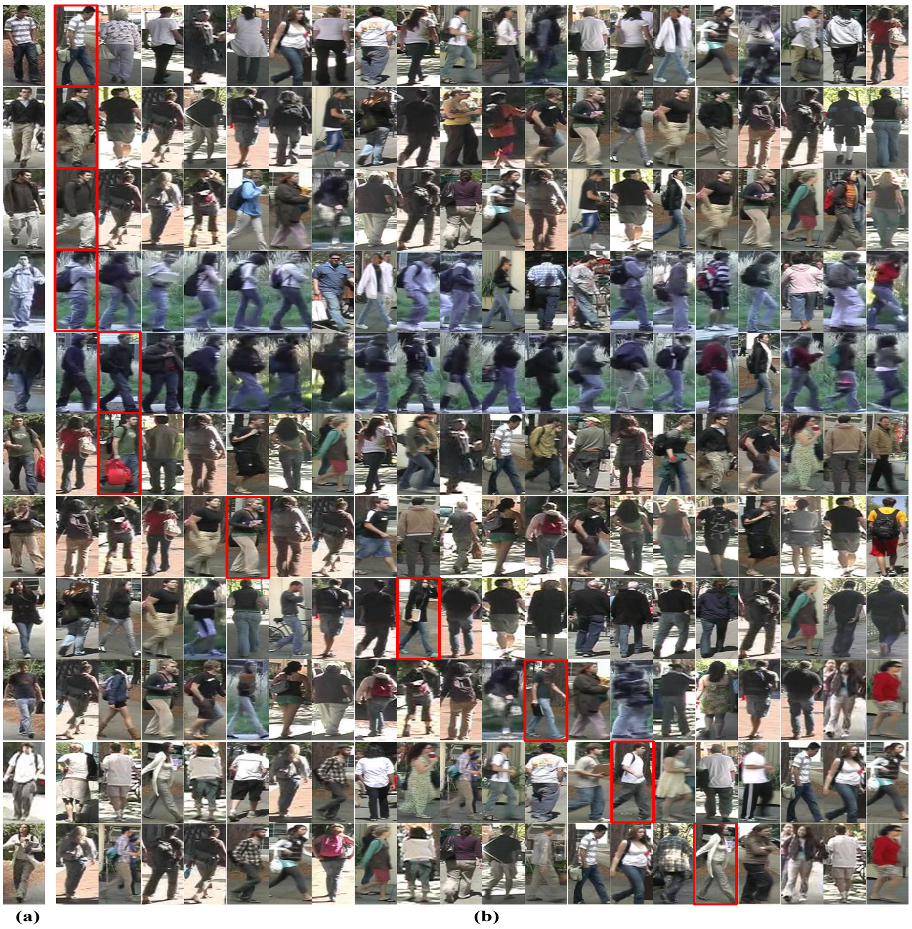

Fig. 6. Examples of the matching results. (a) Reference image; (b) Top 20 results (sorted left to right). The correct matches are circled by red lines.

containing 39 pedestrians and 10 of them walk across cameras. In this case, we only consider pedestrians rather than cyclists or vehicles. D_T and θ_T is set to 10 and 1.0 (radian) respectively. The sequence length N is set to 15, and the $Threshold$ in the sequence based matching framework is set to 0.5. Some examples of tracking results are shown in Figure 10. The tracking accuracy is about 94.9% ($\frac{37}{39}$). All the pedestrians who transfer across cameras are correctly recognized and retain unique labels in the whole videos. In the view of Camera 3, two pedestrians exchange their labels while one is leaving and the other is

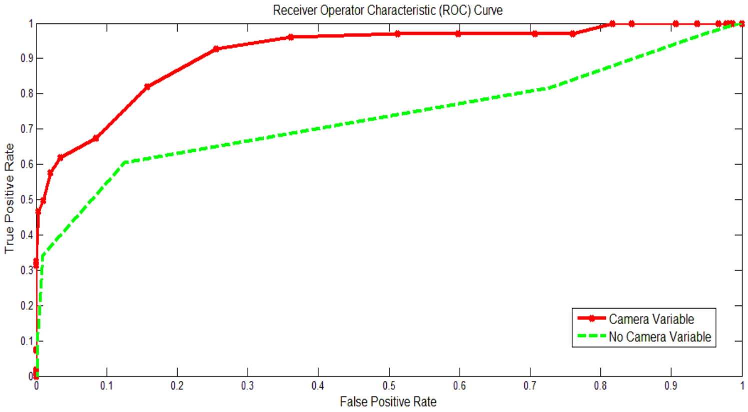

Fig. 7. The receiver operator characteristic curve showing the effectiveness of the categorical variable E^iE^j

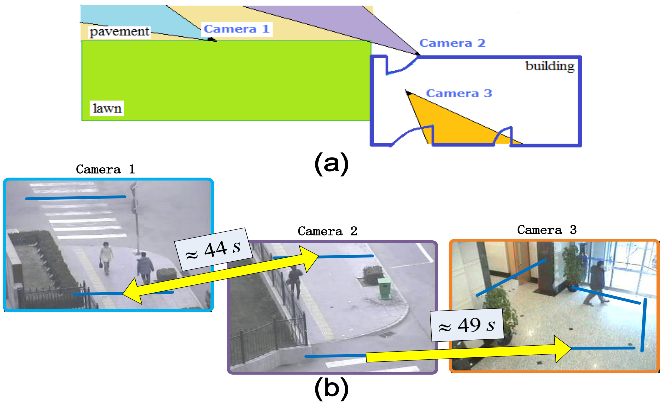

Fig. 8. (a) The layout of the multi-camera system, (b) Learned spatio-temporal cues across cameras

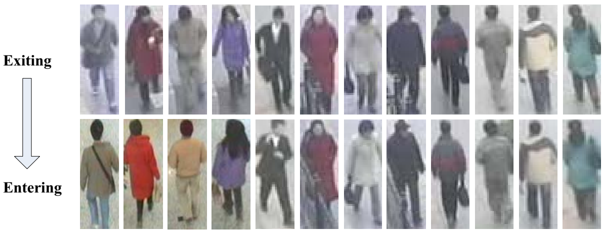

Fig. 9. Some examples used for training. Each column is the same pedestrian in different entry/exit cameras (Column 1-4: from Cam 2 to Cam 3; Column 5-8: from Cam 1 to Cam 2; Column 9-12: from Cam 2 to Cam 1).

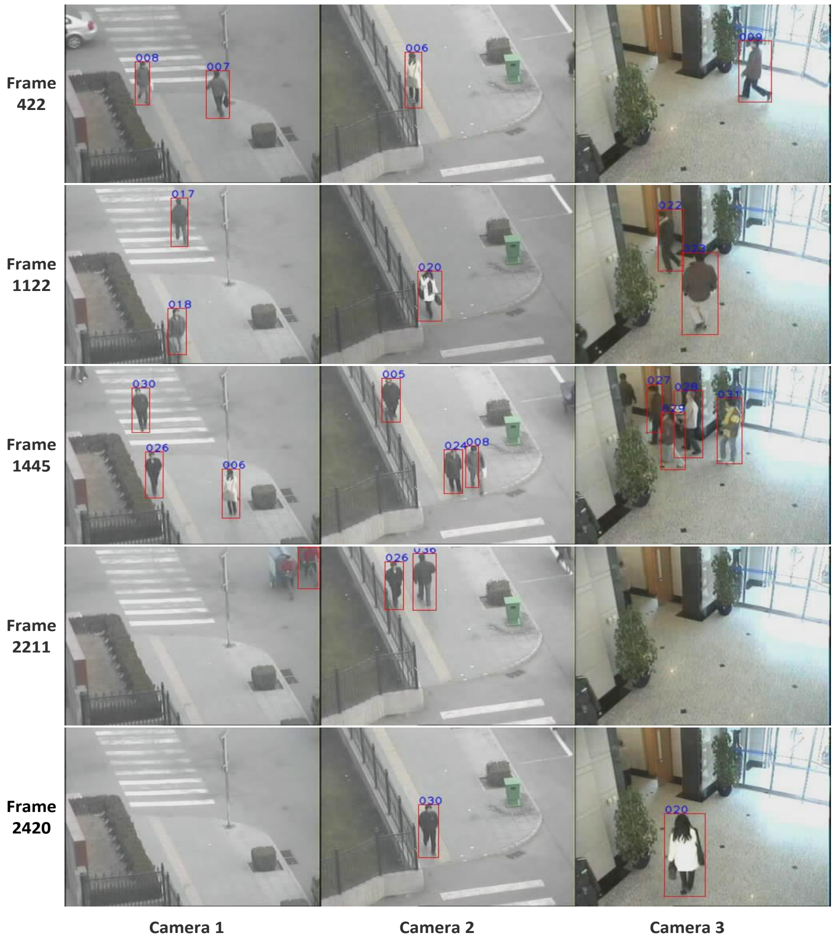

Fig. 10. Examples of tracking results. Note that all pedestrians retain unique labels.

entering almost at the same time and in the same place. This failure can be avoided by combining appearance cues with directional cues in the single camera tracking.

Table 1 shows some outputs of the AdaBoost classifier in the sequence based matching framework, demonstrating the effectiveness of adaptive models. Using the proposed adaptive model, Person 26 (enter) and Person 30 (exit) are impossible to be recognized as the same person due to a very small matching score, while Person 8 (enter) and Person 8 (exit) are successfully matched, although the similarity scores between Person 26 (enter) and Person 30 (exit) are generally larger than that between Person 8 (enter) and Person 8 (exit).

Table 1. Outputs of the AdaBoost classifier in the sequence based matching framework

index	Person 8 (enter) frame 1330-1344 vs. Person 8 (exit) frame 557-571	Person 26 (enter) frame 2187-2201 vs. Person 26 (exit) frame 1511-1525	Person 26 (enter) frame 2187-2201 vs. Person 30 (exit) frame 1621-1635
	$Thre^8 = -15.1423$	$Thre^{26} = -8.0389$	$Thre^{30} = 0.3742$
1	-15.6932	**-7.7642**	-7.2057
2	**-9.3708**	**-1.5252**	-7.2057
3	**-9.0034**	**-1.5252**	-5.4428
4	**-9.0034**	**1.4108**	-7.2057
5	**-9.0034**	**2.0720**	-7.2057
6	**4.3071**	**-5.7348**	-7.2057
7	**4.3071**	**-1.3836**	-3.5291
8	**-8.2373**	**-5.7348**	-7.2057
9	**-8.2373**	**-1.3836**	-8.2373
10	**-8.2373**	**-1.5252**	-9.2772
11	**-8.2373**	**-1.3836**	-8.2373
12	**-8.2373**	**-1.5252**	**4.3071**
13	**-8.2373**	**2.0720**	-7.2057
14	**-8.2373**	**2.0720**	**2.2356**
15	**-8.2373**	**0.8478**	-9.2772
Matching Score	0.93	1.00	0.13

5 Conclusions

In this paper, we have presented a solution to the problem of object tracking across non-overlapping cameras. Unlike previous work, our method deals with the appearance changes differently according to different entry/exit cameras by using a camera variable. Adaptive models are learned and integrated into the sequence based matching framework to draw final conclusions. Experiments have demonstrated that using the camera variable improves the performance, adaptive models are necessary and effective, and the proposed method performs well in tracking multiple objects across non-overlapping cameras. To extend our work, appearance cues can be exploited and integrated into the single camera object tracking, and more distinctive features can be applied to represent the objects when matching across cameras. Future work will focus on object tracking in large-scale multi-camera systems.

Acknowledgement. This work is funded by National Natural Science Foundation of China (Grant No. 61175007), the National Key Technology R&D Program (Grant No. 2012BAH07B01), the National Basic Research Program of China (Grant No. 2012CB316302).

References

1. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1355–1360 (2003)
2. Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S.: Principal axis-based correspondence between multiple cameras for people tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 663–671 (2006)
3. Javed, O., Shafique, K., Shah, M.: Appearance modeling for tracking in multiple non-overlapping cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 26–33 (2005)
4. Prosser, B., Gong, S., Xiang, T.: Multi-camera matching using bi-directional cumulative brightness transfer functions. In: Proceedings of the British Machine Vision Conference (2008)
5. Jeong, K., Jaynes, C.: Object matching in disjoint cameras using a color transfer approach. Machine Vision and Applications 19, 443–455 (2008)
6. Montcalm, T., Boufama, B.: Object inter-camera tracking with non-overlapping views: a new dynamic approach. In: Canadian Conference Computer and Robot Vision, pp. 355–361 (2010)
7. Gray, D., Tao, H.: Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 262–275. Springer, Heidelberg (2008)
8. D'Angelo, A., Dugelay, J.: People re-identification in camera networks based on probabilistic color histograms. In: Visual Information Processing and Communication, SPIE Electronic Imaging, vol. 7882 (2011)
9. Chen, X., Huang, K., Tan, T.: Direction-based stochastic matching for pedestrian recognition in non-overlapping cameras. In: 18th IEEE International Conference on Image Processing, pp. 2065–2068 (2011)
10. Cai, Y., Huang, K., Tan, T.: Human Appearance Matching Across Multiple Non-overlapping Cameras. In: 19th International Conference on Pattern Recognition, pp. 1–4 (2008)
11. Cai, Y., Huang, K., Tan, T.: Matching Tracking Sequences Across Widely Separated Cameras. In: IEEE International Conference on Image Processing, pp. 765–768 (2008)
12. Piccardi, M., Cheng, E.D.: Multi-frame moving object track matching based on an incremental major color spectrum histogram matching algorithm. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, p. 19 (2005)
13. Hu, M., Hu, W., Tan, T.: Tracking people through occlusions. In: International Conference on Pattern Recognition, pp. 724–727 (2004)
14. Birchfield, S., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1158–1163 (2005)

15. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29, 51–59 (1996)
16. Niu, C., Grimson, E.: Recovering non-overlapping network topology using far-field vehicle tracking data. In: International Conference on Pattern Recognition, pp. 944–949 (2006)
17. VIPeR dataset, http://vision.soe.ucsc.edu/?q=node/178
18. Park, U., Jain, A.K., Kitahara, I., Kogure, K., Hagita, N.: ViSE: visual search engine using multiple networked cameras. In: International Conference on Pattern Recognition, pp. 1204–1207 (2006)