Real-World Re-Identification in an Airport Camera Network

Yang Li
Rensselaer Polytechnic Institute, Troy, NY
liy21@rpi.edu

Ziyan Wu
Rensselaer Polytechnic Institute, Troy, NY
ziyan@alum.rpi.edu

Srikrishna Karanam
Rensselaer Polytechnic Institute, Troy, NY
karans3@rpi.edu

Richard J. Radke
Rensselaer Polytechnic Institute, Troy, NY
rjradke@ecse.rpi.edu

ABSTRACT

Human re-identification across non-overlapping fields of view is one of the fundamental problems in video surveillance. While most reported research for this problem is focused on improving the matching rate between pairs of cropped rectangles around humans, the situation is quite different when it comes to creating a re-identification algorithm that operates robustly in the real world. In this paper, we describe an end-to-end system solution of the re-identification problem installed in an airport environment, with a focus on the challenges brought by the real world scenario. We discuss the high-level system design of the video surveillance application, and the issues we encountered during our development and testing. We also describe the algorithm framework for our human re-identification software, and discuss considerations of speed and matching performance. Finally, we report the results of an experiment conducted to illustrate the output of the developed software as well as its feasibility for the airport surveillance task.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Re-identification, camera network, video analytics

ICDSC '14, November 04 - 07 2014, Venezia Mestre, Italy
Copyright 2014 ACM 978-1-4503-2925-5/14/11 ...$15.00.
http://dx.doi.org/10.1145/2659021.2659039 .

1. INTRODUCTION

The academic computer vision research community has expended substantial effort on the problem of human tracking in a camera network with non-overlapping fields of view (e.g., [4, 7, 15]). The most challenging problem is the correct association between tracked people in different camera views. Using appearance features to match images of people in different views is also widely studied as the human re-identification (re-id) problem. Researchers typically approach the re-id problem with an emphasis on feature selection [2, 6, 13] and metric learning [3, 10, 11, 17]. To evaluate and compare the matching performance of the proposed algorithms, results are usually reported on several standard benchmarking datasets agreed upon by the research community.

The story is very different when it comes to re-id in a real-world environment. In addition to addressing a well-defined research problem, i.e., deciding whether two bounding boxes representing humans correspond to the same person, there are many other challenges to building a reliable re-identification application for an actual surveillance system. With respect to hardware, one may need to consider camera installation locations constrained by security limitations of the site, low-quality images from legacy analog cameras equipped in the current network, data storage and transferring strategies, device synchronization, and network bandwidth. With respect to software, the system must operate in near real time and deliver high-quality matching results with few false alarms.

Real-world re-id also differs from academic research on the problem in that most work in the latter case poses the problem as: given a probe image of a person, find the single image of the same person in a gallery of images taken from a different viewpoint. The re-id performance is usually quantified with a curve illustrating the rank n matching rate, i.e., the percentage of probe images that matched correctly with one of the top n images in the gallery set. In the real-world case, each person has multiple images available from being tracked. These can be used to build better descriptors and generate more reliable similarity measurements. Users are unlikely to scroll through pages of candidates, so performance at low ranks (e.g., $n \leq 5$) is critical.

In this paper, we present the system design of a video surveillance solution installed in a real-world airport environment, as well as an algorithm framework for human re-

identification. Our goal is to help airport security officers to detect tagged people of interest in real time. The project involved numerous iterations of on-site tuning, testing and evaluation, and we present the challenges we encountered during its development. We also describe our experiences in moving from academic computer vision algorithm development to "messy" real-world implementation.

2. REAL-WORLD CHALLENGES

Our video surveillance project is centered at a medium-sized airport (Cleveland-Hopkins International Airport, Cleveland, OH, USA). The project goal is to develop an on-site real-time video analytic system to assist Transportation Security Administration (TSA) and airport security staff to track specified people of interest throughout the airport surveillance camera network. We called this task "Tag and Track". This project succeeded a long-term effort at the same site for detecting counterflow though security exit lanes, which shares many of the same challenges discussed here. In this section, we address limitations and challenges we encountered in the real-world system design, installation, running and testing.

2.1 Data Collection, Storage and Transfer

The high-level system design is shown in Figure 1. Unlike traditional surveillance systems, in which camera videos are directly fed into monitoring screens watched by security staff, video data in the airport needs to be transmitted to workstations through a secure high-bandwidth network, and then processed by analytic software.

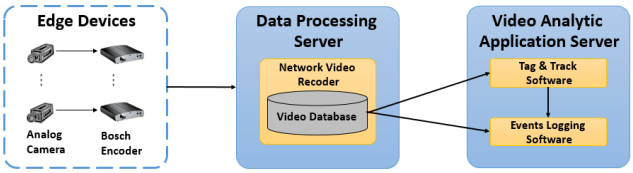

Figure 1: High-level system design of an airport human re-identification solution.

While most academic researchers likely use digital cameras in the lab, many legacy surveillance cameras in long-term installations like airports are still analog. Much of the existing airport surveillance system at CLE is equipped with 4CIF (704× 576) resolution analog cameras, so it was necessary to install video encoders that convert the feeds into digital video data and embed video metadata. We used Bosch video encoders (VideoJet X20/X40 XF E) to convert the video to the H.264 standard. All of the data is then transmitted to a Network Video Recorder (NVR) on a data processing server, which stores the encoded videos and their metadata. Since there is large amount of video data sent to the NVR, it can only store a limited amount of multi-camera data (i.e., about a week of 4CIF 29.97fps video from four cameras).

The developed video analytic software is installed on several separate workstations connected to the data server. These acquire video feeds directly from the encoders and perform tracking and re-identification tasks in real time. All of the data transmissions are via a secure high-bandwidth network, and the whole system is maintained in a local Ethernet (i.e., no access to the outside Internet). Since the accessibility of the surveillance data from the airport is highly restricted, only the workstations connected in the local Ethernet are allowed to query video data from the NVR via proprietary video database tools. Consequently, we had to conduct a large amount of software testing on-site, following a process of developing video analytic algorithms in the lab, testing them on small amounts of recorded data cleared by the airport for our use, and deployed on-site after validation.

To validate the algorithm's on-site performance, we needed to develop a logging program running on the application server to record events. At a certain time every night, the program processes the log files generated by the video analytic program containing the time stamps of the target person and re-id candidates, as well as their locations in each frame. Then it translates the logs into NVR video requests that are sent to the data server. Upon receipt of the requests, a communication channel is set up and the video clip transfer is started. When the transfer is done, the logging program labels the file with the time stamps of the corresponding event. Event logging is a crucial component of the system since it enables the accurate evaluation of the system performance and relieves the storage pressure on the data server. However, it also adds a considerable amount of data transfer load to the network, and must be carefully timed to avoid interfering with actual events, or querying the NVR at the same time by different processes. The recorded events are reviewed by security officers, and brought back to the lab for analysis roughly every month.

2.2 Time Stamp Synchronization

To evaluate the system performance, security officers review the extracted video clips of re-identification events. To ensure accurate video extraction from the NVR, it is critical that the time stamps for all cameras, encoders and servers are well-synchronized. However, during software testing we found that the video encoders send out metadata, following the Real-time Transport Protocol (RTP), containing time stamps with a small drift compared to the time stamps in the cameras, data server and application server. Since the NVR maps this time stamp to the server's system time and uses it to create a list of videos to be retrieved, this drift led to accumulated time differences, so that the time intervals in the extracted video clips did not correspond to the requested intervals. This is actually not uncommon for video encoders and IP cameras, and is difficult to fix from the customer side.

However, since we observed that the drift of the video encoders was stable and repeatable (i.e., a constant drift each day), but disappeared after resetting the encoder, we found it was easier to program the logging software to calculate an estimated drift according to the dates of the latest reset of the encoders, and compensate by adjusting the time intervals of the requested video clips from the NVR. While this was an easy fix, it required resetting all encoders after a certain amount of time to remove accumulated uncompensated drift. As a longer-term fix, a protocol was added to the NVR to actively initiate time syncing with all the encoders every hour, effectively resetting the encoders constantly.

2.3 Image Quality

We observed that several of the legacy analog cameras in the system contain serious noise and may not maintain

focus over time. Figure 2 illustrates several sample images. It can also be observed that illumination conditions vary from camera to camera. Even for the same camera, the illumination can change throughout the day or with respect to weather conditions. The reflective decorative tile floor also makes foreground detection more difficult. Finally, the videos also contain periodic temporal jitter that seriously affects tracking algorithms. We discussed our solution to this problem in Wu et al. [16].

The heavy traffic environment in the airport makes it even harder to detect and track people. The worst situation occurs when many passengers get off a plane, causing a very crowded scene in which people can be passed or occluded by others. In such cases, it is extremely difficult to maintain accurate trajectories for each person. We will discuss our tracking and re-id strategy in Section 3.

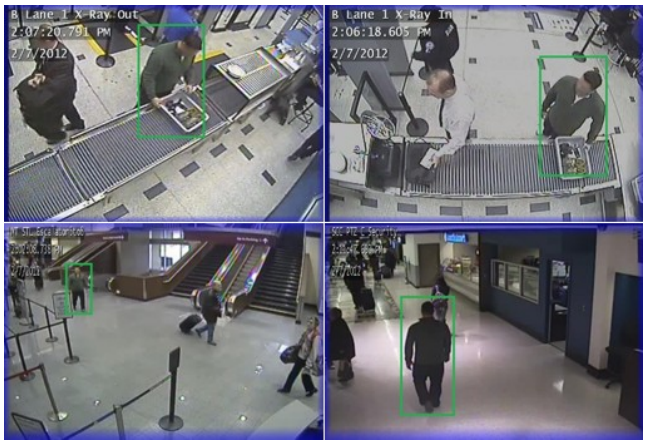

Figure 2: Sample images from airport camera videos.

2.4 Camera Position

Like most public surveillance systems, the camera network in the airport cannot cover the whole area of interest. In fact, the fields of view are mostly non-overlapping with large "blind" regions. On the other hand, the movements of humans in an airport are more unpredictable compared with other surveillance scenarios, such as traffic flow monitoring. For example, in most views there are no pre-defined routes or directions for people; after walking out of one view, people can walk back into the same view, while the algorithm may expect to detect the person in a different camera. There are many entrances and exits that are not covered by cameras, so that people can appear or disappear from the monitoring area with high uncertainty; people may stay for long periods or even change clothes out of the view of any camera, which can cause the estimation of their motion based on a fixed appearance or a transit-time model to fail. This issue is especially problematic around security screening checkpoints, which have restrictions on camera placement due to privacy and law enforcement regulations. Re-identification in this scenario is extremely challenging. Finally, unlike the standard datasets used to evaluate re-id algorithms, which contain images taken from cameras whose optical axes have a small angle with the ground plane, in the airport environments, the angle between the camera optical axis and the floor is usually large ($\sim 45^\circ$), causing serious perspective effects (see Figure 2).

3. ALGORITHM FRAMEWORK

In this section, we describe our algorithm framework for the airport surveillance re-identification problem, emphasizing feasibility considerations for the real-world environment. The goal is to provide reliable re-id candidates corresponding to a tagged target person in real time. Figure 3 illustrates the major computer vision steps in the process.

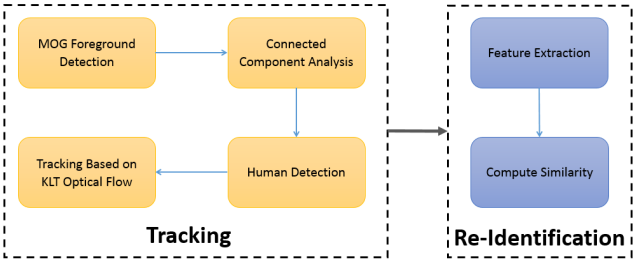

Figure 3: Our human re-identification algorithm framework.

3.1 Detection and Tracking

We begin with foreground detection using the mixture of Gaussians (MoG) method [14], followed by connected component analysis to group foreground pixels into blobs. The bounding box of each blob is considered as the region of interest (ROI). The ROI is then fed into a real-time pedestrian detection algorithm; we adopted the aggregated channel features framework of Dollár et al. [5]. Specifically, a boosted decision tree classifier is used in conjunction with a sliding window approach. The classifier is trained using 3000 ground truth pedestrian images (forming the positive sample set) and randomly sampled background images (forming the negative sample set) from the airport videos. For the purposes of training, we formed multi-scale feature pyramids by aggregating six quantized gradient orientations, L, U, and V color channels, and normalized gradient magnitudes into a ten-channel feature vector, computed over each scale. The result is a set of candidate detections of different sizes inside each ROI blob. Once the order is received to find a tagged person, human detection starts to run in all frames of each camera, since new humans may enter the scene at any time.

At the same time, the bounding boxes of tracked people from the previous frame are propagated to the current frame and their estimated locations updated. We detect low-level FAST corners [12] inside each tracked bounding box and track them with the KLT tracker [9]. The estimated position of each bounding box in the current frame is produced by pushing the bounding box in the previous frame forward by the average of the motion vectors of the detected features. To obtain a more reliable motion vector, we remove the scene feature points using the background scene classifier described in [16].

Finally, the two sets of candidate bounding boxes in the current frame are merged. We compute the intersection of each new human detection with the bounding boxes propagated from the previous frame, and find the maximum ratio r between this area of intersection and the area of the smaller bounding box. If this ratio is above a suitable threshold

(we used 0.8 in our experiments), the new human detection is associated with the corresponding human in the previous frame. Otherwise, a new track is initialized with the new detected bounding box. Propagated bounding boxes matching no human detection in the current frame are also retained if their aspect ratio and location in the frame are reasonable. Figure 4 illustrates the idea.

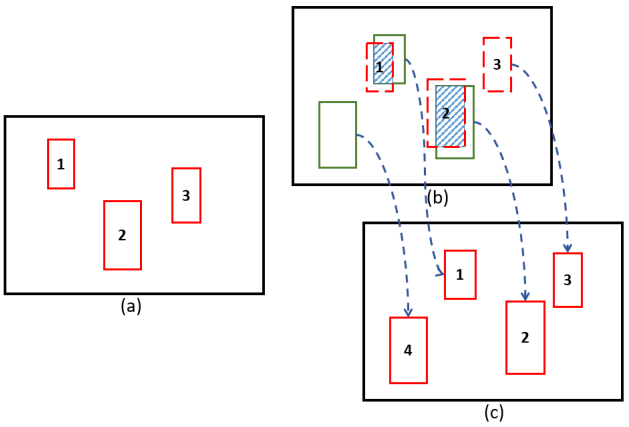

Figure 4: (a) Bounding boxes from previous frame. (b) Dashed bounding boxes (red) are propagated from previous frame using feature detection and optical flow; solid boxes (green) are new candidates generated by the human detector. (c) Final bounding boxes in current frame created by merging the two detections.

3.2 Re-identification

The re-identification process has two key steps. First, we must extract a feature descriptor from each candidate detection. Second, given a pair of descriptors $\mathbf{X}_{\text{target}}$ and $\mathbf{X}_j$ (one from the tagged target and the other from the j^{th} candidate detection), we must find an appropriate similarity score

$$s_j = f(\mathbf{X}_{\text{target}}, \mathbf{X}_j) \tag{1}$$

Then, by ranking the similarity scores $\{s_j, j = 1, \ldots, n\}$ in each frame, we generate an ordered list of "preferred" candidates that are shown to the user.

Feature detection for re-id in real-world scenarios is challenging, especially given the relatively small and low-quality target and candidate images. Common descriptors like SIFT [8] and SURF [1] are unsuitable for the task. Instead, we found low-level features such as color and texture histograms to be more effective and efficient. In particular, we adopted the method described by Gray and Tao [6]. The image is divided into 6 horizontal strips. Inside each strip, 16-bin histograms are computed over 8 color channels (RGB, HSV, and CbCr) and 19 texture channels (including the response of 13 Schmid filters and 6 Gabor filters). By concatenating all the histograms we get a 2592-dimensional feature vector for each candidate. We found it was important to rectify the candidate sub-images based on simple camera calibration information to remove perspective distortion prior to feature extraction. In the future, we also plan to incorporate radiometric and color calibration across the cameras.

The next step is find a metric to accurately quantify similarity. Many metric learning techniques have been proposed for re-id [3, 10, 11, 17]. In our algorithm, we applied the rankSVM method [11] to maximize the norm of a weight vector $\mathbf{W}$ subject to the constraints that if

$$\mathbf{d}_{\text{same}} = |\mathbf{x}_\mathbf{j}^\mathbf{i} - \mathbf{x}_\mathbf{k}^\mathbf{i}|$$

is the absolute difference of descriptors for two images of the same person i, and

$$\mathbf{d}_{\text{diff}} = |\mathbf{x}_\mathbf{j}^\mathbf{i} - \mathbf{x}_\mathbf{k}^\mathbf{l}|$$

is the absolute difference of descriptors for images of two different people i and l, then

$$\mathbf{W}^\top \mathbf{d}_{\text{same}} < \mathbf{W}^\top \mathbf{d}_{\text{diff}}$$

for all possible pairs from same and different people. We trained the weight vector using manually annotated ground truth data extracted from the airport videos, which includes images from around 150 people. The re-id distance function between two descriptors is thus computed as

$$f(\mathbf{X}_{\text{target}}, \mathbf{X}_j) = \mathbf{W}^\top |\mathbf{X}_{\text{target}} - \mathbf{X}_j|$$

Most current human re-id algorithms [3, 6, 10, 17] are focused on the "single-shot" problem; that is, it is assumed that each person only has one image available to compute the similarity score. This assumption is mainly motivated by the limited data contained in public re-id benchmark datasets. However, in the real-world scenario, there is a sequence of images available for each tracked person, which is the "multi-shot" case. Let $\{\mathbf{X}_1^t, \mathbf{X}_2^t, \cdots, \mathbf{X}_n^t\}$ be feature descriptors collected for the target person, and $\{\mathbf{X}_1^c, \mathbf{X}_2^c, \cdots, \mathbf{X}_m^c\}$ be the feature descriptors collected for a candidate person. We then calculate an accumulated similarity score s_a for the candidate as

$$s_a = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} f(\mathbf{X}_i^c, \mathbf{X}_j^t) \tag{2}$$

We expect this accumulated similarity score to give a more accurate measure of similarity than the single-shot version. The target person is tagged manually by a security officer, so we assume that the n shots of the target are clear and reliable. However, the automatically generated bounding boxes for the tracked candidate might not be highly accurate; e.g., there could be occlusions or drifted tracks. We observed that the bounding boxes generated by the human detection algorithm are more likely to correspond to clear, well-posed human images, and are usually maintained for a small window of consecutive frames. Thus, we modified (2) to choose the k consecutive frames of the candidate with the highest total similarity score, such that at least one bounding box comes from the human detection algorithm:

$$s_a = \frac{1}{kn} \max_i \sum_{i}^{i+k-1} \sum_{j=1}^{n} f(\mathbf{X}_i^c, \mathbf{X}_j^t) \tag{3}$$

In our experiment, we used $k = 5$. We also note that multi-shot information can be used to train a discriminative model of the target person on-line, improving re-id performance.

3.3 Algorithm Discussion

When developing the algorithm, we had to consider requirements for both speed and performance. The algorithm needs to be fast enough to process multiple cameras in real time, and at the same time, find the person of interest with high confidence.

Human detection is the most time-consuming step; our implementation is close to real time (around 15 fps on our videos). By filtering out ROIs with small sizes or impossible locations, and only analyzing viable ones, we highly reduced the computational cost to around 100 fps. While the process of training the re-id weighting vector is time-consuming, this is done offline. The on-line re-id process is extremely fast since it only involves a vector inner product. There is enough spare computational power in our system to consider online re-id learning algorithms, such as updating representative feature vectors after the same person is confirmed in another view, or discriminative model training.

We found it was important to consider the "big picture" of how good the results of each sub-process needed to be in order to result in a confident re-id judgement, instead of trying to squeeze the best performance out of every algorithm at the possible cost of speed. For example, we know the MoG foreground detection is likely to fail when the surrounding illumination changes, but this can be mitigated later in the pipeline by the human detection step. In fact, we need a relatively sensitive foreground detection algorithm to make sure we won't miss any people in the detection stage (resulting in many false alarms that are rejected later). Similarly, there is no state-of-the-art tracking algorithm that can process multiple streams of airport-quality videos with high accuracy in real time. The tracking algorithm we applied may fail when a candidate person is occluded, several trackers become focused on the same person, or the bounding box drifts onto a different person. However, all we care about is generating a sufficient number of reliable candidates for the multi-shot algorithm; occluded or poor-quality rectangles will simply never rise to the top of the rank-ordered candidate list.

4. EXPERIMENTAL RESULTS

Here, we report the results of one experiment using real-world airport videos to demonstrate the overall re-id performance of our system. We chose three cameras located in the area between the parking garage and the airport terminal. Sample images of the camera views are shown in Figures 5a-c. People coming from the parking garage will be seen first in camera A. They then proceed to camera B (at which point there are stairs and elevators enabling them to enter or exit the environment). If they continue to move forward, they will appear in camera C and move into the terminal.

In each experiment, we tagged a person in camera A and then tracked him or her until they disappeared from the field of view. The target's feature vectors are extracted from the tracking frames. After the target leaves camera A, we begin to detect and track all the candidates in camera B and C for a 5-minute period, as shown in Figures 5b-c. We can see that the tracking task is very challenging in camera B, because of the distorted view angle and the crowded scene. However, as discussed in Section 3.3, as long as the person is successfully detected and tracked for a short distance, the program can still make a reliable judgment. One example re-id result is shown in Figure 5d. We only display the top 5 candidates to the user in ascending order of similarity score. In this example, the target person is ranked second in camera B and third in camera C. Although detection and tracking in camera B is more challenging, the viewpoint (the person's back), is more similar to the tagged viewpoint, so the re-id results are better in camera B than in camera C.

We repeated the experiment for 40 targets in camera A over 12 hours of video, selecting candidates who appeared in all three views. The results are presented in Table 1, which reports the percentage of target images that matched correctly with one of the top n images in the detected candidates. We found that 70% of the targets were found in camera B and 65% of the candidates were found in camera C within the top 5 automatically generated results. We use rank 5 as a rule of thumb for assessing performance, since at this point a human should be able to easily decide whether or not the candidate is a correct match.

	Rank 1	Rank 5	Rank 10	Rank 20
Camera B	37.5%	70%	92.5%	100%
Camera C	30%	65%	87.5%	100%

Table 1: Re-identification results for the experiment.

5. CONCLUSIONS

We discussed several practical challenges in implementing a real-time re-identification solution in a mid-sized airport, which might not be typically considered by academic researchers, and presented initial results from our algorithm framework tailored to this setting. However, there is still much work to be done, both in our specific environment, and more generally to make academic re-id research more closely match real-world scenarios.

With respect to our specific environment, we are only at the beginning of our implementation and testing of the on-site re-id system, following a successful deployment of a system for real-time detection of counterflow through exit lanes described elsewhere [16]. From a computer vision point of view, we plan to incorporate more robust algorithms for incorporating the estimated poses of the target and candidate into the descriptor comparison, refining the multi-shot re-id strategy, and learning subject-discriminative features on-line. We also plan to incorporate spatio-temporal prior knowledge about the camera arrangement to cull unlikely re-id candidates among widely distributed cameras. The most pressing practical problems include designing a robust, crash-resistant software architecture that can run for days at a time, and creating an intuitive user interface that allows the user to easily retain possible matches and reject others. We also had the unique opportunity to design a new re-id testbed at the airport, containing higher-quality digital cameras, positioned as carefully as possible within security and power constraints to capture a complex branching re-id scenario (i.e., a passenger exiting the security checkpoint can enter one of three concourses, after spending an unknown time in a shopping area). We expect this new testbed to generate further challenges from both the research and practical perspectives.

Acknowledgements: This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. Thanks to Michael Young and Jim Spriggs for supplying the airport video data. Thanks to Deanna Beirne and Rick Moore for helping to set up and maintain the described system.

(a) (b) (c) (d)

Figure 5: Sample results from the airport human re-identification system. (a) Tagging the person of interest in camera A, (b) Tracking in camera B, (c) Tracking in camera C, (d) Re-identification results (green boxes indicate correct candidates).

6. REFERENCES

[1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[2] L. Bazzani, M. Cristani, and V. Murino. Symmetry-driven accumulation of local features for human characterization and re-identification. *CVIU*, 117(2):130–144, 2013.

[3] J. Blitzer, K. Q. Weinberger, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005.

[4] K.-W. Chen, C.-C. Lai, P.-J. Lee, C.-S. Chen, and Y.-P. Hung. Adaptive learning for target tracking and true linking discovering across multiple non-overlapping cameras. *Multimedia*, 13(4):625–638, 2011.

[5] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *PAMI*, 2014.

[6] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008.

[7] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *CVPR*, 2005.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*, 1981.

[10] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[11] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary. Person re-identification by support vector ranking. In *BMVC*, 2010.

[12] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 32(1):105–119, 2010.

[13] W. R. Schwartz and L. S. Davis. Learning discriminative appearance-based models using partial least squares. In *SIBGRAPI*, 2009.

[14] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[15] X. Wang, K. Tieu, and W. E. L. Grimson. Correspondence-free activity analysis and scene modeling in multiple camera views. *PAMI*, 32(1):56–71, 2010.

[16] Z. Wu and R. J. Radke. Improving counterflow detection in dense crowds with scene features. *Pattern Recognition Letters*, 44:152–160, 2014.

[17] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011.