A multi-view pedestrian tracking method in an uncalibrated camera network

Domonkos Varga[1,2] Tamás Szirányi[1,2] Attila Kiss[1] László Spórás[1]
László Havasi[1]
[1]MTA SZTAKI
Kende u. 13-17, 1111 Budapest, Hungary

[2]Budapest University of Technology and Economics
Műegyetem rkp. 3, 1111 Budapest, Hungary

{varga.domonkos, sziranyi.tamas, kiss.attila, sporas.laszlo, havasi.laszlo}@sztaki.mta.hu

Abstract

Combining multiple observation views has proven beneficial for pedestrian tracking. In this paper, we present a methodology for tracking pedestrians in an uncalibrated multi-view camera network. Using a set of color and infrared cameras, we can accurately tracking pedestrians for a general scene configuration. We design an algorithmic framework that can be generalized to an arbitrary number of cameras. A novel pedestrian detection algorithm based on Center-symmetric Local Binary Patterns is integrated into the proposed system. In our experiments the common field of view of two neighboring cameras was about 30%. The system improves upon existing systems in the following ways: (1) The system registers partially overlapping camera-views automatically and does not require any manual input. (2) The system reaches the state-of-the-art performance when the common field of view of any two cameras is low and successfully integrates optical and infrared cameras. Our experiments also demonstrate that the proposed architecture is able to provide robust, real-time input to a video surveillance system. Our system was tested in a multi-view, outdoor environment with uncalibrated cameras.

1. Introduction

Multi-view pedestrian tracking for video surveillance received a lot of attention in recent years, which is motivated by security applications and the development of intelligent robots. Compared to single view, multiple views and different modalities of the same scene can be used to recover information that might be missing in a particular view or modality. Multiple target tracking usually contains two main steps: the first step is the detection of objects of interest and the second is their temporal linkage from frame to frame.

1.1. Related work

There is extensive literature on multi-camera detection and tracking algorithms. An extensive review on tracking and multi-view tracking is beyond the scope of this paper. We refer readers to comprehensive surveys [36], [20] for more details about existing trackers. In this section, we review only the works related to our method.

There are a few single camera tracking algorithms that take scene priors into account to improve the tracking accuracy, however, these methods are not straightforward to extend to multi-camera distributed tracking scenarios [30], [31]. Cai and Aggarwal [2] extended a single-camera tracking system. They switched another camera when the system predicts that the current camera will no longer have a satisfactory view of the subject.

Recently, tracking by detection algorithms have been gaining popularity. Existing multiple camera tracking algorithms do not discriminatively model the multi-view appearance in an online manner. Detection based tracking algorithms obtain object hypotheses by applying an object detector to images. The detector is learned off-line from labeled training data. Given detection responses generated by the detector, the tracking algorithm needs to retrieve the real objects among those responses and set ID for each of them in every frame.

Orwell et al. [27] presented a tracking algorithm to track multiple objects in multiple views using color tracking. They modeled the connected blobs obtained from background subtraction using color histogram techniques and use them to match and track objects. Khan et al. [17] presented a novel planar homography constraint to robustly determine locations on the ground plane corresponding to the feet of the pedestrian. To find tracks they obtained feet regions over a window of frames and stack them creating a space time volume. Fernández-Caballero et al. [9] presented a thermal-infrared pedestrian detection system under different outdoor environmental conditions. It was introduced an algorithm for pedestrian ROI extraction in infrared video based on both thermal and motion information. Chrysostomou et al. [3] proposed a multi-view optimisation process to get the best geometric condition for surveillance. [12] is based on the detected pedestrian to build a grouping model for crowd analysis.

Utasi et al. [34] developed a probabilistic approach on multiple calibrated camera views. The presence of people in the scene are approximated by a population of cylinder objects in the 3D world coordinate system, which is a realization of a Marked Point Process. The observation model is obtained from the projection of the pixels of the motion masks in the different camera frames to the ground plane and to other parallel planes with different height. Kiss et al. [18] developed a real-time pedestrian tracking based on leg detection for cases of different ground-plane height. First, the foreground mask is filtered in order to find pixels relevant to detecting position of people. Then spatially coherent pixels are collected to form one primitive from them. The authors filtered pixels possibly corresponding to feet, which are called candidate pixels. These pixels are covered with ellipses, these can be back-projected to cones in scene space.

1.2. Contributions

The major contributions of this paper are listed as follows:

1. The paper presents a new methodology for tracking pedestrians in a multi-camera network. In this network, the Fields of View (FOV) of two arbitrary cameras is not greater than 30 %.

2. The present work do not consider any predefined geometrical constraints nor object or scale pre-definitions for calculating inter-camera transformation. The method is based on co-motion statistical analysis [33] for different modalities (infra and other cameras) what looks like an efficient solution for views having no any identical image-features.

3. We present our novel pedestrian detector based on Multi-scale Center-symmetric Local Binary Pattern. A new feature extraction pipeline is introduced which mainly captures contour information.

4. Based on registering the results of different pedestrian detectors, we present a new multi-view tracking algorithm using König's theorem and the Hungarian method.

2. The overview of our system

The overview of our system is presented in Figure 1. The Video/camera module obtaines the frames of the cameras, corrects the distortions, and provides the synchronized frames for the whole system. Using these frames, the Pedestrian detection module provides the coordinates of the pedestrians bounding box in each frame. The Trajectory module consists of two parts. The first part requires the synchronized camera frames for the image registration. The second part supplies the trajectories of the pedestrians. In the rest of this chapter we describe the algorithms which work in the individual modules.

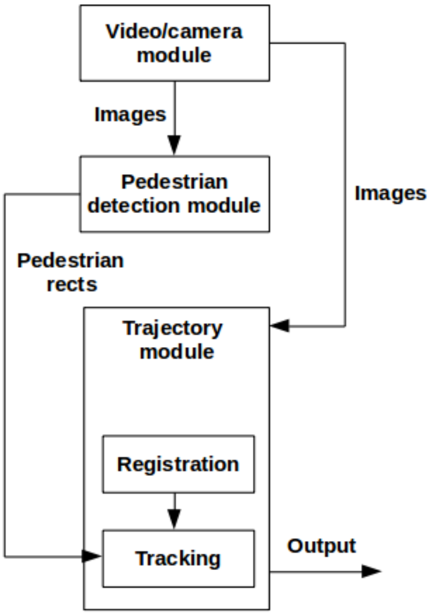

Figure 1. The overview of the proposed system.

2.1. Registration

The camera distortions is an important error source in such systems where we want to correspond accurately the images of many cameras. In our case it is essential to solve this problem in the registration, in the detection and in the tracking too. Lens distortion is a complex field, and a lot of approaches have been developed to address this problem [13], [35], [1].

After correcting the lens distortion the matching is achieved by calculation of co-motion statistics. The robust algorithm we describe in this subsection finds point correspondences in two images without searching for any structures and without the need for tracking continuous motion. In our experiments the common field of view of two neighboring cameras was about 30%. The detailed description of the algorithm can be found in the work of Szlávik, Havasi, and Szirányi [33]. Here we outline only the main steps of the algorithm:

1. Detecting motion: record the point coordinates where motion is detected. The motion blobs are extracted by using running-average background subtraction with large β, deleting the irrelevant parts by using the previous image I_{k-1} as a reference:
$$I_k(x,y) = \beta I_k(x,y) - (1-\beta)I_{k-1}(x,y), 0 < \beta < 1. \quad (1)$$
2. Updating local and remote statistical maps. In order to find correspondences between two images, we analyze the dynamics of the scene by co-motion (concurrent motion) statistics. For each pixel of the two images, a local statistical map and a remote statistical map are generated.
3. Extracting candidate point-pairs from the statistical maps. Then it can be considered as a transition matrix of an ergodic regular Markov chain with states [33]. According to the Frobenius-Perron theorem [32], such a Markov chain has a unique stationary distribution.
4. Rejecting points that are not relevant because they lie outside the common field-of-view. Point-pairs, in which both of points are from the overlapping views are assumed to be the inliers while any other point-pairs are the outliers. To perform it, Bayes-decision algorithm was implemented.
5. Fine-tuning point correspondences by minimizing the reprojection error between the candidate point-pairs. An iterative technique is used to refine the point placements based on Levenberg-Marquardt iteration.
6. Aligning the images from the separate cameras.

Using the cited [32], [33] algorithms we are able to determine if two cameras have common area of interest. With the help of co-motion statistics we can determine the plane of the ground. In the calculation we assume that the ground is approximately flat. Under these conditions and knowing the location of the ground, ground-homography can be calculated between any two cameras that have a flat common area of interest on the ground.

If we imagine the camera network as a graph, a vertex represents a camera, and there is a link between two vertices, if ground-homography exists between the two cameras represented by the vertices. We select a camera or vertex randomly - called reference camera or reference vertex. We determine the spanning tree of the graph. With the help of the spanning tree we can calculate the pathes to the reference vertex from all the other vertices. A path gives us the product of the ground-homographies that is necessary to obtain the view in the reference camera.

2.2. Pedestrian detection

Our pedestrian detection system scan the video frames all relevant positions and scales to detect a pedestrian. A feature component encodes the visual appearance of the pedestrian, while the classifier component determines for each sliding window independently whether it contains a pedestrian or not (Figure 2).

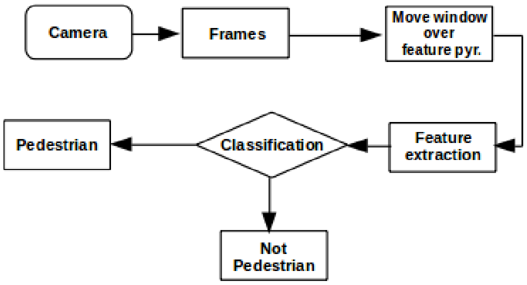

Figure 2. Architecture of the pedestrian detection module.

To train our system, we gathered a set of 13,500 greyscale sample images of pedestrians as positive training examples, together with their left-right reflections. The positive examples have been aligned and scaled to the dimensions 128 × 64. The images of the pedestrians were taken from public pedestrian datasets [16], [6] and from our surveillance and traffic videos. We made a database of negative samples too, which consists of 16,000 non-pedestrian images. In order to improve the performance we put 7,000 vertical structures like poles, trees or street signs to the negative samples. The vertical structures are common false positive detections in pedestrian detection.

Feature is the key in pedestrian detection and other pattern recognition problems. A good feature is able to obtain discriminative information between the pedestrian class and others, and it is stable with respect to intra-class variances. Our goal was to develop a feature that is discriminative enough both for RGB images and thermal images because our test environment integrates optical and infrared cameras. We describe the feature extraction method in the followings.

2.2.1 Multi-scale Center-symmetric Local Binary Pattern Operator

Local Binary Pattern (LBP) is a simple, but very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number [29]. The original LBP operator labels the pixels of an image by thresholding the 3-by-3 neighborhood of each pixel with central pixel value and the result is taken as a binary number. A histogram of the labeled image $f_l(x,y)$ can be calculated:

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\},\ i = 0, ..., n-1, \qquad (2)$$

where n is the number of different labels and

$$I\{A\} = \begin{cases} 1 & \text{if A is true} \\ 0 & \text{if A is false.} \end{cases} \qquad (3)$$

The LBP operator was later extended to many other variations. In this paper we are interested in the so-called Center-symmetric Local Binary Pattern (CS-LBP) [14]. In CS-LBP, pixel values are not compared to the center pixel but to the opposing pixel symmetrically with respect to the center pixel. We can see that for 8 neighbors, LBP produces 2^8 different binary patterns, whereas for CS-LBP this number is only 2^4.

The idea of Multi-scale Center-symmetric Local Binary Pattern is based on the simple principle of varying the radius R of the CS-LBP label and combining the resulting histograms. Developing the idea of [14], the neighborhood is described with two parameters $P, R = \{R_1, R_2, ..., R_{n_R}\}$, where n_R is the number of radii utilized in the process of calculation. Each pixel in Multi-scale CS-LBP image is described with n_R values. The multi-scale CS-LBP histogram for different values of $R = \{R_1, R_2, ..., R_{n_R}\}$ can be determined by summing $H^{(1)}, H^{(2)}, ..., H^{(n_R)}$ vectors:

$$H = \sum_{i=1}^{n_R} H^{(i)}. \qquad (4)$$

2.2.2 Feature extraction

In this paragraph, we introduce the implementation details of the feature extraction. We believe that contour is the most useful information for pedestrian detection, and our feature extraction method mainly captures the contour. The key steps of feature extraction are as follows. All the listed steps resulted in significant improvement in the classification performance.

1. We normalize the gray-level of the input image to reduce the illumination variance in different images. After the gray-level normalization, all input images have gray-level ranging from 0 to 1.

2. We obtain 4 layers of the input image in the following way: first, we compute the gradient magnitude of each pixel of the input gray-scale image (detection window), then we repeat this calculation three times on the previous derivative image. Considering the speed of the calculation, we compute an approximation of the gradients using Sobel operator.

3. The detection window and each of the four layers of the detection window are split into equally sized overlapping blocks. The rate of overlapping is 50 %. In our case, the size of the detection window is 64×128 and the size of the blocks is 16×16.

4. We take the detection window and the multi-scale CS-LBP histograms ($P = 8$, $R_1 = 1$, $R_2 = 2$, $R_3 = 3$, $n_R = 3$) are extracted from each block independently. Let v_i be the unnormalized descriptor of the ith block, f be the descriptor of the detection window. We obtain f in the following way:

   - $f = [v_1, v_2, ..., v_N]$;
   - l_1-norm, $f \leftarrow f/\sqrt{(\| f \|_1 + \epsilon)}$;

5. We take each layers one after the other and the multi-scale CS-LBP histograms are extracted from each block independently. Let $v_{i,j}$ be the unnormalized descriptor of the ith block in the jth layer, g_j be the descriptor of the jth layer. We obtain g_j in the following way:

   - $g_j = [v_{1,j}, v_{2,j}, ..., v_{N,j}]$;
   - l_1-norm, $g_j \leftarrow g_j/\sqrt{(\| g_j \|_1 + \epsilon)}$;

6. We obtain the feature vector of the detection window in the following way:

$$F = f + \sum_{j=1}^{4} \frac{1}{j+1} g_j \qquad (5)$$

We can see that the feature vector of the original image (f in Eq. 5) mainly captures the contours, the feature vector of the $4th$ layer (g_4 in Eq. 5) mainly captures the detailed textures or cluttered background, the rests capture special edges or textures. That is why the weights of the layers in Eq. 5 have descending coefficients. There are various parameter configurations that can be chosen in order to optimize the performance of the above described feature based detection approach. We chose the parameters of the feature extraction with respect to our experimental results.

The overall length of the feature vector for a 128×64 detection window is $7 \times 15 \times 16 = 1680$ because each window is represented by 7×15 blocks. Experiments on the INRIA pedestrian dataset show that the proposed multi-scale CS-LBP feature with support vector machine with radial basis function performs well.

2.2.3 Feature representation

In many applications such as video surveillance, detection speed is as important as accuracy. A standard pipeline for performing multi-scale detection is to create a densely sampled image pyramid then the detection system scans all images of the pyramid to detect a pedestrian. In order to accelerate the scanning process, we define a feature pyramid using a standard image pyramid.

We obtain the four layers of an image of the standard pyramid as described in the previous subsection. The Multiscale CS-LBP operator ($P = 8$, $R_1 = 1$, $R_2 = 2$, $R_3 = 3$, $n_R = 3$) is applied to the image and its four layers. In this way we correspond five values to each pixel of the image. An image of the standard pyramid can be substituted by an $(W - 2 \cdot R_3) \times (H - 2 \cdot R_3) \times 5$ array where W stands for the width of the image and H is the height of the image. Using the feature pyramid derived from a standard image pyramid, the time of the feature extraction and thereby the scanning process can be reduced.

2.3. Tracking

The next phase of our method is to register the results of the different pedestrian detectors on each cameras. Because of the overlapping of the different image planes a single object can be detected on multiple cameras. Hence if we project all the detected objects from all the cameras onto the top-view image there might be multiple points belonging to the same object if it is detected on multiple image planes (this time we use the foot nodes of the objects). The distance of the corresponding point pairs on the top-view image is a function of the accuracy of the detecors in use. The more accurate the output of the detectors the smaller the distance between the corresponding point pairs. We set an order between the points of the top-view image by topologically sweeping an arrangement. Sweeping a vertex set in the Euclidean plane with a straight line is a well-known algorithmic paradigm in computational geometry. Edelsbrunner and Guidas showed [7] that if we use a topological line that is not necessarily straight we can get other advantages. They showed that an arrangement of n lines can be swept over $O\left(n^2\right)$ time and $O\left(n\right)$ space by such a line. Furthermore during this process each element (i.e. vertex, edge or region) is visited once in a consistent ordering. We use this ordering in our method as well. Hence we can register the corresponding points from different views by using this ordering and a radius set by the user. This radius defines a region around each node in the top-view image. This region is the only area where we search for corresponding points from different views. After this step we have exactly one top-view image point belonging to a single object. We can increase the efficiency of the algorithm if we choose a double sweep in two different (possibly orthogonal) directions in two dimensions.

The next step is to assign these registered top-view objects with the previously stored and later updated objects in the system and refresh our knowledge. Because of the continuous processing the amount of the objects stored in the system is not equal with the amount of the top-view objects of the following state. For example an object might leave or enter the scene in the next state. Hence the position of the objects might change because they are allowed to move. So if the user set the radius correctly our algorithm will efficiently handle this case. Then let us given a matrix A where the indices of the columns of the matrix are assigned to the objects stored in the system and the indices of the rows of the matrix are assigned to the objects currently present on the scene. The values in a_{ij} are distances between the (possibly) new object i and the stored object j. If there is a threshold on the value of a_{ij}, it can't be larger than the radius. Hence if it is too large then we set the value of a_{ij} to the value of the radius. We try to find exactly one cell from each row and column of the matrix such as the sum of the values of these cells is minimal. To find a solution like this we applied the famous Hungarian method of Harold W. Kuhn [19].

If there are more stored objects in the system than the number of incoming objects then our matrix has more columns than rows. In this case we have to insert extra rows into the matrix because we need a square matrix. We have to insert extra columns into our matrix if there are more incoming objects than stored ones of course. The value of the cells of these extra rows or columns is the fixed radius.

It is not trivial how to choose a minimal number of rows and columns in a matrix to cover all the zero values in the matrix. But in a graph theoretical point of view this problem can be well approximated. We define a bipartite graph $G = (C, R; E)$ on the matrix. The vertices of the set C belong to the columns of the matrix and and the set R is for the rows of the matrix. Connect the vertices $c_i \in C$ and $r_j \in R$ with an edge if $a_{ij} = 0$ in the matrix A. In this bipartite graph we determine a maximum matching. By a theorem of Dénes König [23] the size of this matching is equal to the minimal number of cover nodes in this graph:

Theorem 1 (König)**.** *[23] Let A be an $m \times n$ 0–1 matrix. Then the term rank of A equals the minimum number of lines required to cover all zeroes in A.*

In this maximum matching each edge belongs to exactly one row and one column. We have to decide if we need the column or the row in our covering of the zero elements. Our method search for the minimal number of rows and columns to cover all the zero elements in the matrix.

If we apply the algortihmic proof of the König theorem to find a maximum matching in this graph we can find minimum covering vertex set as well. The proof of this statement can be found in the monograph of András Frank [11] with the proof of König's theorem as well.

We store the history of different image points for the registered objects. Hence the positions can be set more accurate for example by the Kalman filter. If the object is behind an obstacle and we do not have enough information about its position then we can approximate it.

3. Experimental results: detection and tracking

In the experimental section we evaluated two main parts of the proposed system that have comparable numerical results:

1. the efficiency and speed of the pedestrian detection,
2. the tracking accuracy by calculating the precision and recall values.

Using test sequences, we analyzed our system. The test sequences include various people moving through the scene with other moving objects including. Pedestrian detection was determined to be a success if the appropriately sized bounding box encapsulated the pedestrian in the scene. All experiments were performed online.

Pedestrian detection: Figure 3 shows the detection rate versus false positive per-image (FPPI) for the proposed pedestrian detector and seven other detectors. The nine systems we compare include Dalal and Trigg's HOG+SVM system [4], Lie et al. HOG+Adaboost system [22], Papageorgiou et al. Haar+SVM system [28], Monteiro et al. Haar+AdaBoost system [26], a HOG+IKSVM system [24], a PHOG+HIKSVM system [24], LatSvm detector [8], ChnFtrs [5] and our proposed system (Multi-scale CS-LBP + SVM). From the results we can see that our method has a powerful and discriminative feature that is superior to others. It could reduce the false detections significantly.

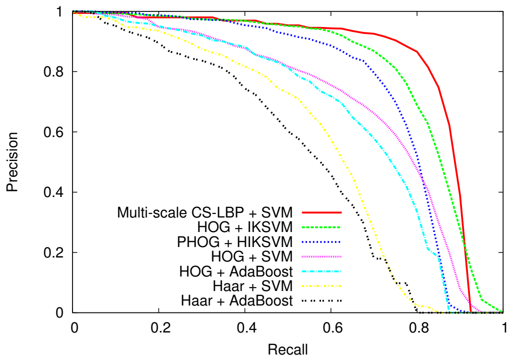

Figure 3. Detection rate versus false positive per-image (FPPI) curves for pedestrian detectors. 4×4 is the step size and 1.09 is the scale factor of the sliding-window detection.

The speed comparison of the seven systems is presented in Table 1. We measured the speed at 640×480 resolution and the accuracy at 1 FPPI (false positive per image).

Table 1. Speed comparison of several pedestrian detection systems. The resolution is 640×480.

Method	Speed
Haar+SVM [28]	56.7 fps
Haar+AdaBoost [26]	70.1 fps
PHOG+HIKSVM [24]	2.1 fps
HOG+IKSVM [24]	2.55 fps
HOG+Adaboost [22]	22.3 fps
HOG+SVM [4]	2.45 fps
LatSvm [8]	0.78 fps
ChnFtrs [5] (no infra)	88.7 fps
ours (optical or infra)	38.1 fps

As we mentioned, the test environment contains optical and infrared cameras too. That is why the pedestrian detection system have to work well on thermal images. Our presented feature extraction method captures mainly gradient and edge information, some texture and scale information. This properties enable to the whole system the sufficient performance both on RGB and thermal images. Some of the other state-of-the-art methods listed in Table 1 were not able to give appropriate performance on thermal images [e.g. Dollar], while they slightly outperforms our method in accuracy for visible color channels.

Tracking: We could not find similar cases in the literature with wide-baseline infra/optical multiview *uncalibrated* arrangement. However, to evaluate our method's efficiency we compared it to mostly state-of-the-art tracking methods in *calibrated* multiview cameras. We compared our system to six other methods referred to as POM [10], 3DMPP [34], M2 tracker [25], Tensor Voting tracker [15], Relaxation tracker [21] and ParFit [18]. Table 2 shows the results of the other algorithms and the results of our system in the last row. As it can be found in Table 2, our system's efficiency is not far from that of the best calibrated systems. We evaluated our system in an *outdoor, real environment* (see Figure 5) while the other systems were tested in *indoor environment*. The top-view image of our measurement arrangement can be seen in Figure 5. The error of center location of target is calculated for every frame as

$$e_c = ||\mathbf{K}_t - \mathbf{K}_{groundtrue}||, \tag{6}$$

where $\mathbf{K}_t$ is the target location at the tth frame, and $\mathbf{K}_{groundtrue}$ stands for the groundtrue location. If the target's center location error is greater than 0.25 m in a frame, we say that the tracker fails [18], [34]. Using the temporal sequence of center location error, we are able to determine the ROC curve of the system. The ROC curve of the proposed system can be seen in Figure 4.

Figure 6 and Figure 7 show some tracking results. In Figure 6 four consecutive frames of three different cameras (two RGB and one thermal) can be seen. These three cameras form in this case an uncalibrated camera network.

Table 2. Precision/recall values for tracking comparing our uncalibrated method to the state-of-the-art well-calibrated and uncalibrated methods.

	Precision	Recall
POM (calibrated) [10]	87.2 %	95.56 %
3DMPP (calibrated) [34]	97.5 %	95.5 %
ParFit (calibrated) [18]	90.66 %	95.61%
M2 tracker (calibrated) [25]	83.9 %	90.4 %
Tensor Voting (uncalibrated) [15]	77.6 %	80.2 %
Relaxation (calibrated) [21]	70.1 %	72.3 %
ours (uncalibrated)	84.3 %	95.13 %

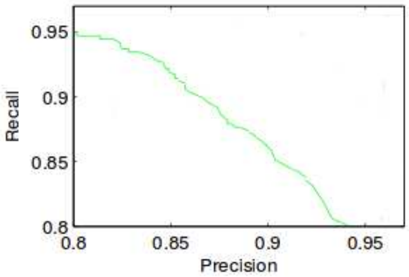

Figure 4. ROC curve of tracking measured in function of detection threshold.

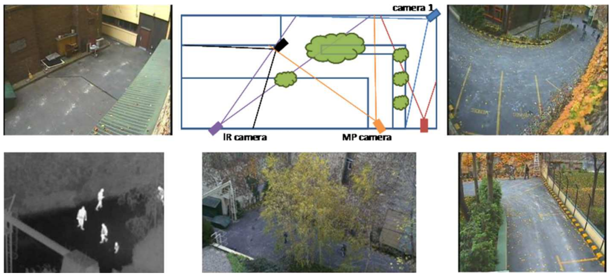

Figure 5. The test environment of our measurement.

Figure 7 shows the determined tracjectory of the walking pedestrians and the projected views.

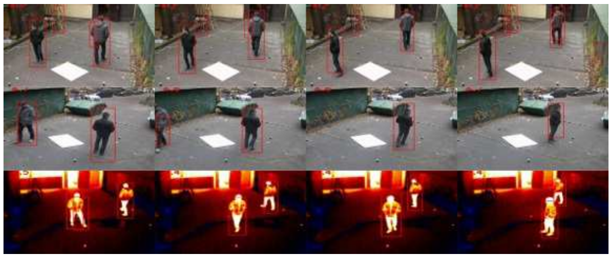

Figure 6. Frames of three different cameras and the detection results (the third camera was an infra one).

4. Conclusion and Outlook

In this paper we presented a novel algorithmic framework for real-time detecting and tracking pedestrians in a multi-camera network. This framework is able to register the images of different cameras using co-motion statistics. The framework was also used to train pedestrian detector for real scenes. Using the results of the registration and the pedestrian detector, we design a real-time tracking method based on König's theorem and the Hungarian method that performs well in multi-view, outdoor environment with uncalibrated cameras. The evaluation demonstrates we achieve high accuracy if the common field of view of two neighboring cameras is about 30%. Our experiments also demonstrate that the pedestrian detector can provide robust input for a tracking framework and it is able to work on different modalities.

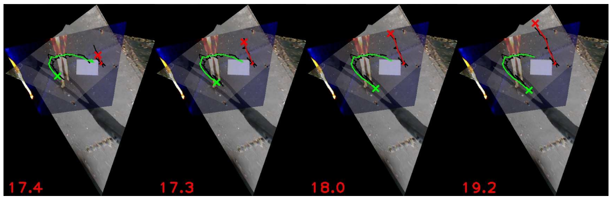

Figure 7. Some tracking results in a test sequence.

Acknowledgments

This work has been supported by the EU FP7 Programme (FP7-SEC-2011-1) No. 285320 (PROACTIVE project). The research was also partially supported by the Hungarian Scientific Research Fund (No. OTKA 106374).

References

[1] D. G. Bailey. A new approach to lens distortion correction. *Proceedings Image and Vision Computing New Zealand 2002*, pages 59–64, 2002.

[2] Q. Cai and J. Aggarwal. Tracking human motion using multiple cameras. In *ICPR*, volume 3, pages 68–68. IEEE, 1996.

[3] D. Chrysostomou, G. C. Sirakoulis, and A. Gasteratos. A bio-inspired multi-camera system for dynamic crowd analysis. *Pattern Recognition Letters*, 44:141–151, 2014.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[5] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, volume 2, page 7. Citeseer, 2010.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[7] E. Edelsbrunner and L. J. Guidas. Topologically sweeping an arrangement. In *ACM Symposium on Theory of computing*, pages 389–403, 1986.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[9] A. Fernández-Caballero, M. T. López, and J. Serrano-Cuerda. Thermal-infrared pedestrian roi extraction through thermal and motion information fusion. *Sensors*, 14(4):6666–6676, 2014.

[10] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, 2008.

[11] A. Frank. Connections in combinatorial optimization. *Discrete Applied Mathematics*, 160(12):1875, 2012.

[12] W. Ge, R. T. Collins, and R. B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):1003–1016, 2012.

[13] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with center of distortion estimation. *PAMI, IEEE Tr. on*, 29(8):1309–1321, 2007.

[14] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing*, pages 58–69. Springer, 2006.

[15] J. Kang, I. Cohen, and G. Medioni. Continuous multi-views tracking using tensor voting. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 181–186. IEEE, 2002.

[16] C. G. Keller, M. Enzweiler, and D. M. Gavrila. A new benchmark for stereo-based pedestrian detection. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 691–696. IEEE, 2011.

[17] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Computer Vision–ECCV 2006*, pages 133–146. Springer, 2006.

[18] Á. Kiss and T. Szirányi. Localizing people in multi-view environment using height map reconstruction in real-time. *Pattern Recognition Letters*, 34(16):2135–2143, 2013.

[19] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel. A survey of appearance models in visual object tracking. *ACM Tr. on Intelligent Systems and Technology (TIST)*, 4(4):58, 2013.

[21] Y. Li, A. Hilton, and J. Illingworth. A relaxation algorithm for real-time multiple view 3d-tracking. *Image and vision computing*, 20(12):841–859, 2002.

[22] G. Lie, G. Ping-shu, Z. Yi-bing, Z. Ming-heng, and L. Lin-hui. Pedestrian detection based on hog features optimized by gentle adaboost in roi. *Journal of Convergence Information Technology*, 8(2), 2013.

[23] L. Lovász and M. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[24] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, pages 1–8, 2008.

[25] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[26] G. Monteiro, P. Peixoto, and U. Nunes. Vision-based pedestrian detection using haar-like features. *Robotica*, 24:46–50, 2006.

[27] J. Orwell, P. Remagnino, and G. Jones. Multi-camera colour tracking. In *Visual Surveillance, 1999. 2nd IEEE Workshop on*, pages 14–21. IEEE, 1999.

[28] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *ICCV*, volume 8(2), pages 555–562, 1998.

[29] M. Pietikäinen. Image analysis with local binary patterns. In *Image Analysis*, pages 115–118. Springer, 2005.

[30] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1472–1485, 2009.

[31] S. Santhoshkumar, S. Karthikeyan, and B. Manjunath. Robust multiple object tracking by detection with interacting markov chain monte carlo. In *Proc. ICIP*, 2013.

[32] Z. Szlávik, L. Havasi, and T. Szirányi. Image matching based on co-motion statistics. In *International Symposium on 3D Data Processing, Visualization and Transmission*, pages 584–591, 2004.

[33] Z. Szlávik, T. Szirányi, and L. Havasi. Stochastic view registration of overlapping cameras based on arbitrary motion. *Image Processing, IEEE Tr.*, 16(3):710–720, 2007.

[34] A. Utasi and C. Benedek. A 3-d marked point process model for multi-view people detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3385–3392. IEEE, 2011.

[35] J. Wang, F. Shi, J. Zhang, and Y. Liu. A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607–615, 2008.

[36] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.