Homography Based Multiple Camera Detection and Tracking of People in a Dense Crowd

Ran Eshel and Yael Moses
Efi Arazi School of Computer Science,
The Interdisciplinary Center, Herzliya 46150, Israel

Abstract

Tracking people in a dense crowd is a challenging problem for a single camera tracker due to occlusions and extensive motion that make human segmentation difficult. In this paper we suggest a method for simultaneously tracking all people in a dense crowd using a set of cameras with overlapping fields of view. To overcome occlusions, the cameras are placed at a high elevation and only people's heads are tracked. Head detection is still difficult since each foreground region may consist of multiple subjects. By combining data from several views, height information is extracted and used for head segmentation. The head tops, which are regarded as 2D patches at various heights, are detected by applying intensity correlation to aligned frames from the different cameras. The detected head tops are then tracked using common assumptions on motion direction and velocity. The method was tested on sequences in indoor and outdoor environments under challenging illumination conditions. It was successful in tracking up to 21 people walking in a small area (2.5 people per m^2), in spite of severe and persistent occlusions.

1. Introduction

In this paper we present a new method for tracking multiple people in a dense crowd by combining information from a set of cameras overlooking the same scene. People tracking is a well-studied problem in computer vision, mainly, but not exclusively, for surveillance applications. Most existing methods are based on single camera tracking, which does not cope well with crowded scenes such as the one shown in Fig. 1. For example, trackers based on motion segmentation (e.g., [25]) are expected to fall short since large parts of the image are moving. Other methods, based on shape, appearance of the body or its parts, combination of motion and shape, or features learned by training [13, 14, 18, 21], are expected to encounter difficulties since the body parts are not isolated, and may be significantly occluded.

To avoid occlusion as much as possible, we only detect heads. We place a set of cameras at a high elevation, from which the heads are almost always visible. Head segmentation using a single image is still extremely difficult, since in a dense crowd, people are often merged into large foreground blobs (see Fig. 4). In particular, lower body parts, which may be used as shape cues, remain occluded. To overcome this problem, our method merges information from a set of static cameras with overlapping fields of view. Four views of the same scene are shown in Fig. 1. The cameras are assumed to be synchronized and partially calibrated.

We rely on the assumption that the head is the highest region of the body. A head top forms a 2D blob on the plane parallel to the floor at the person's height. The set of frames taken from different views at the same time step is used to detect such blobs. For each height, the foreground images from all views (each may be a blob containing many people) are transformed using a planar homography [4] to align the projection of the plane at that height. Intensity correlation in the set of transformed frames is used to detect the candidate blobs. In Fig. 2 we demonstrate this process on a scene with a single person. Repeating this correlation for a set of heights produces 2D blobs at various heights that are candidate head tops. By projecting these blobs to the floor, multiple detections of the same person at different heights can be removed. At the end of this phase, for each time step we obtain the centers of the candidate head tops projected to the floor of a reference sequence.

In the next phase of our algorithm, the detected head top centers are combined into tracks. At the first level of tracking, atomic tracks are detected using conservative assumptions on the expected trajectory, such as consistency of motion direction and velocity. At the second level, atomic tracks are combined into longer tracks using a score which reflects the likelihood that the two tracks belong to the same trajectory. Finally, a score function based on the length of the trajectory and on the consistency of its motion is used to detect false positive trajectories and filter them out.

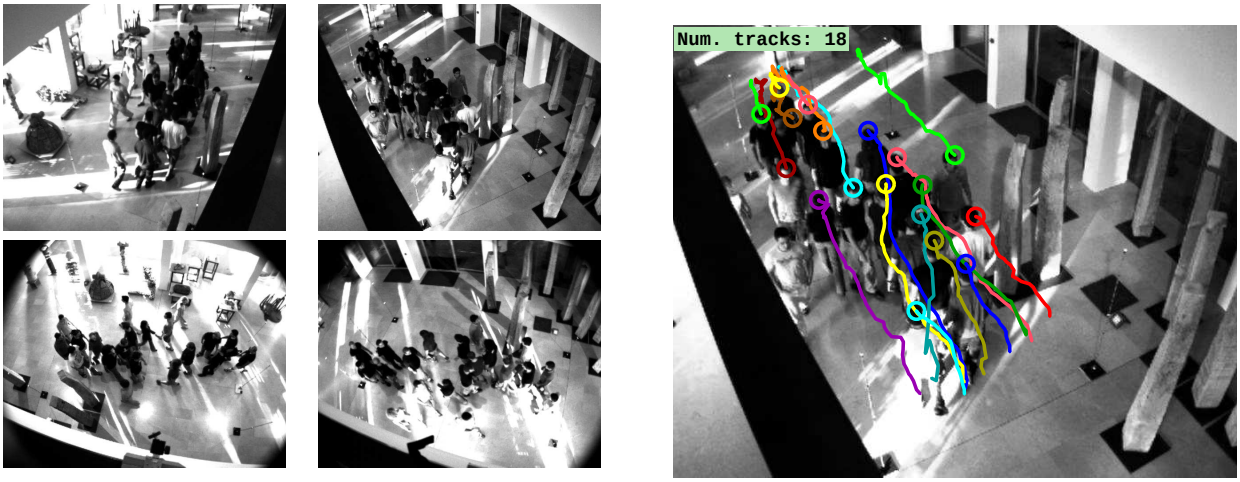


Figure 1. Four views of the same scene, with tracking result on the reference frame

The method was tested on indoor and outdoor sequences with challenging lighting conditions, and with up to 21 people walking in a small area (2.5 people per m^2). It overcomes hard challenges of tracking people: severe and persistent occlusions, subjects with non-standard body shape (e.g., a person carrying a suitcase or a backpack), people wearing similar clothing, shadows and reflections on the floor, highly varied illumination within the scene, and poor image contrast. To overcome these challenges, our method integrates information from a set of cameras. Although the use of multiple cameras with overlapping fields of view may be considered too costly, the economics of camera prices and the proliferation of cameras in many crowded arenas is likely to make this scenario ever more common.

The rest of the paper is organized as follows. In the next section we present a short review of previous work. Sec. 3 describes the head top detection phase and the tracking phase. Our experiments are presented in Sec. 4, and we discuss our results and suggest future research directions in Sec. 5.

2. Related Work

Most methods for tracking several people, who may be close together, are based on a single camera. An example based on combining local and global shape and appearance, is given in Leibe et al. [12]. Wu et al. [22] suggest a learning system for defining local and global features for detecting and tracking partially occluded people. Zhao et al. [25] suggest using shape information for segmenting and tracking people walking in close proximity, and hence appearing in the same foreground blob. Statistical methods were also used to segment people in similar settings [7, 19]. These and other single camera methods appear to be inadequate for handling dense crowds as considered in this paper. For example, the state-of-the-art single view tracking system developed by Wu, Zhao & Nevatia was reported to be inapplicable under the challenging density and illumination conditions considered in this paper.[1]

One method that operates in a similar density to ours is presented in Rabaud and Belongie [17]. Their system is designed to count moving objects by segmenting tracked features. However, they provide only a rough count of people and do not perform full tracking.

Multiple cameras are mostly used in tracking for extending the limited viewing area of a single camera. In this case, single camera tracking is performed for each camera, and the responsibility of tracking a given subject is transfered from one camera to another [1, 3, 5, 16]. Krumm et al. [10] use pairs of cameras to resolve ambiguity using 3D stereo information. Mittal & Davis [14] suggested a higher level of collaboration between cameras, where foreground blob ambiguity is solved by matching regions along epipolar lines. Both methods are based on background subtraction, and hence are limited when a dense crowd is considered. Fleuret et al. [6] suggest using four cameras in order to overcome occlusions. They combine a generative model with dynamic programming for tracking up to six people.

The method most similar to ours for detecting people from multiple cameras was proposed by Khan & Shah [8]. They use a homography transformation to align the foreground of the floor plane from images taken from a set of cameras with overlapping fields of view. Their method, however, is likely to cause many false positive errors due to shadows, reflections on the floor, and the density of the crowd. This is because they rely on foreground and background values rather than on intensity values, and consider the floor plane rather than planes at different heights above the floor. We demonstrate these problems in Fig. 4**b**, where we align only foreground regions of the sequences at a given

[1] Personal communication.

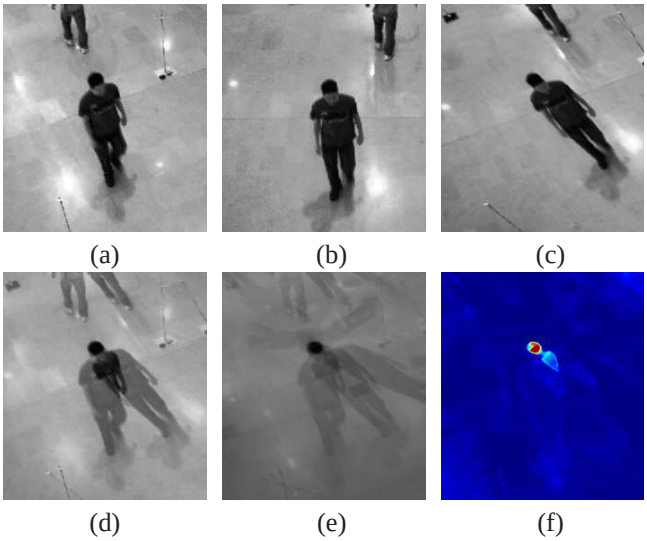


Figure 2. π^{178}-mapping allows to detect a patch at this hight. (a), (b) Two views of the same person. (c) Image (b) aligned to the plane π^{178} of image (a). (d) Image (c) overlaid on image (a). (e) Overlay of plane transformation of additional images. (f) Variance map of the hyper-pixels of (e) color coded such that red corresponds to low variance.

height. A related method, which considers planes at multiple heights, was recently suggested by Khan et. al. [9], but used for 3D shape recovery of non-occluded objects, and not for tracking.

Tracking methods can also be classified according to how motion assumptions are used. Comprehensive reviews on tracking can be found in [20, 23]. Statistical tracking methods use prediction of object shape and location relying on previous measurements to assist in object detection, label objects along the sequence and improve efficiency of the computation (e.g., [2, 7, 15, 24]). We employ heuristic predictions that rely on previous measurements using assumptions on the motion of people. These predictions are used to reduce ambiguity in correspondence between frames and to overcome missing measurements. Detection of candidate features is performed in all frames, under the assumption that a new trajectory can appear at any time and place.

3. The Method

We assume a set of cameras overlooking the same scene, where all head tops are visible. The cameras are assumed to be synchronized and partially calibrated. The partial calibration of the cameras consists of the homography of 3 planes parallel to the floor between each pair of cameras. In Sec. 4 a method of obtaining such calibration and synchronization is suggested.

Initially, head top centers and their heights are detected (each represented by a single feature point), and projected to the floor. These feature points are then tracked to recover the trajectories of people's motion, and filtered to remove false positives.

3.1. Head Top Detection

The head top is defined as the highest 2D patch of the person. The detection of candidate head tops is based on *co-temporal* frames, that is, frames taken from different sequences at the same time. Since we assume synchronized sequences, co-temporal frames are well defined. Fig. 4 shows intermediate results of the method described below.

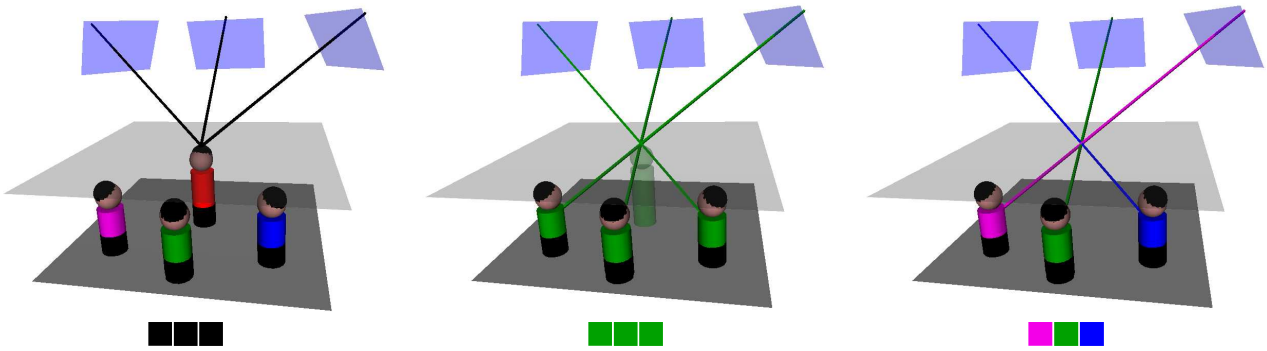
2D patch detection: To detect a 2D patch visible in a set of co-temporal frames, we use the known observation that images of a planar surface are related by a homography transformation. When a homography transformation is applied to images of an arbitrary 3D scene, the points that correspond to the plane will align, while the rest of the points will not. This idea is demonstrated in Fig. 2 for a single person at height 178 cm.

Consider n synchronized cameras overlooking the same scene. Let S_i be the sequence taken by camera i, with S_1 serving as the reference sequence. Assume that π^0 is the floor plane defined by $Z = 0$. Let π^h be a plane in the 3D scene parallel to the image floor at height h. A *π-mapping* between an image and a reference image is defined as the homography that aligns the projection of points on the plane π in the two images. For a plane π^h and sequences S_i and S_1, it is given by the 3×3 homography matrix $A^h_{i,1}$. Using the three known homography matrices given by the partial calibration, $A^{h1}_{i,1}$, $A^{h2}_{i,1}$ and $A^{h3}_{i,1}$, the homography matrices $A^h_{i,1}$ can be computed for any height h.

Consider $S_1(t)$, a frame of the reference sequence in time t. To detect the set of pixels in $S_1(t)$ that are projections of a 2D patch at height h, the co-temporal set of n frames is used. Each of the frames is aligned to the sequence S_1, using the homography given by the matrix $A^h_{i,1}$. Let $S_i(t)$ be a frame from sequence i taken at time t. Let $p \in S_i(t)$, and let $I_i(p)$ be its intensity. A *hyper-pixel* is defined as an $n \times 1$ vector $\bar{q}^h$ consisting of the set of intensities that are π^h-mapped to $q \in S_1(t)$. The π^h-mapping of the point $p \in S_i(t)$ to a point q in frame $S_1(t)$ is given by $q = A^h_{i,1} p_i$. The inverse transformation, $p_i = A^h_{1,i} q$, allows to compute $\bar{q}^h$:

$$\bar{q}^h = \begin{pmatrix} I_1(q) \\ I_2(p_2) \\ . \\ . \\ I_n(p_n) \end{pmatrix} = \begin{pmatrix} I_1(q) \\ I_2(A^h_{1,2} q) \\ . \\ . \\ I_n(A^h_{1,n} q) \end{pmatrix}.$$

The hyper-pixel $\bar{q}^h$ is computed for each pixel $q \in S_1(t)$. Highly correlated intensities within a hyper-pixel indicate that the pixel is a projection of a point on the considered



(a) Low variance (b) Low variance (c) High variance

Figure 3. After applying the plane transformation which corresponds to the imaginary plane in the scene, the hyper-pixel of the aligned images will contain the marked rays. (a) A 3D point at the plane height is detected where a person is present. (b) A false positive detection occurs due to accidental projections of points from different people. This will only happen if all points coincidentally have the same color. (c) In the more common case, points belonging to different objects have different colors. This results in high hyper-pixel intensity variance, which prevents false positive detection.

plane π^h. A low correlation can be expected for other points provided that the scene is not homogeneous in color. Using hyper-pixel intensity variance, we obtain a set of pixels that are likely to be projections of points on the plane π^h. Simple clustering, using double threshold hysteresis on these pixels and a rough estimation of the head top size (in pixels), can be used for detecting candidate 2D patches on the plane π^h. If a blob is larger than the expected size of a head top, a situation that may occur in extremely dense crowds, the blob is split into several appropriately sized blobs using K-means clustering. The number of clusters is determined by the blob size and the expected head size. The centers of the 2D patches are then used for further processing.

Note that the background is likely to be homogeneous because the floor and the walls in an indoor scene, or the ground in an outdoor scene, are expected to have similar colors. We therefore align only the foreground regions from each of the co-temporal frames. We use a naive background subtraction algorithm, which subtracts each frame from a single background frame (taken when the scene was empty). The results of this simple background subtraction are sufficient for our method.

Detecting the highest 2D patch: The process of detecting 2D patches is repeated for a set $H = \{h_1, ..., h_n\}$ of expected heights of people. The set is taken at a resolution of 5 cm. We assume that the head tops are visible to all cameras. It follows that at this stage of our algorithm, all head tops are detected as 2D patches at one of the considered heights. However, each person can be detected as 2D patches at several heights. The goal here is to recover only the highest patch for each person.

To do so, we compute the foot location of each of the 2D patches as would appear in the reference sequence. The foot location is assumed to be the orthogonal projection of a 2D patch at a given height h to the floor. The projection is computed using a homography transformation from the reference sequence to itself. The homography aligns the location of each point on the plane π^h in the reference image with the location of its projection to the plane π^0 in the same image. For each height $h_i \in H$, the homography transformation that maps the projection of the plane π^{h_i} to the floor of sequence S_1 is given by the 3×3 homography matrix B^{h_i}. These matrices can be computed on the basis of the partial calibration assumption of our system. For head top center $q \in S_1(t)$, detected at height h, the projection to the floor of S_1 is given by $B^{h_i}\, q$.

For each floor location, a single 2D patch is chosen. If more than one patch is projected to roughly the same foot location, the highest one is chosen, and the rest are ignored. This provides, in addition to detection, an estimation of the detected person's height, which can later assist in tracking.

Expected 'phantoms': Phantoms typically occur when people are dressed in similar colors, and the crowd is dense. As a result, portions of the scene may be homogeneous in color, and accidental intensity correlation of aligned frames may be detected as head tops. Fig. 3**b** illustrates such accidental intensity correlation where the plane alignment will correlate non-corresponding pixels, originating from different people who happen to be wearing the same color. Phantoms can also affect the detection of real people walking in the scene: the head of a phantom can be just above a real head, causing it to be removed since it is not the highest patch above the foot location. The probability of detecting phantoms can be reduced by increasing the number of cameras (see Sec. 4.3). We remove phantoms in the tracking phase. Phantom removal can be further improved by applying single camera human body segmentation methods, but this is beyond the scope of this paper.

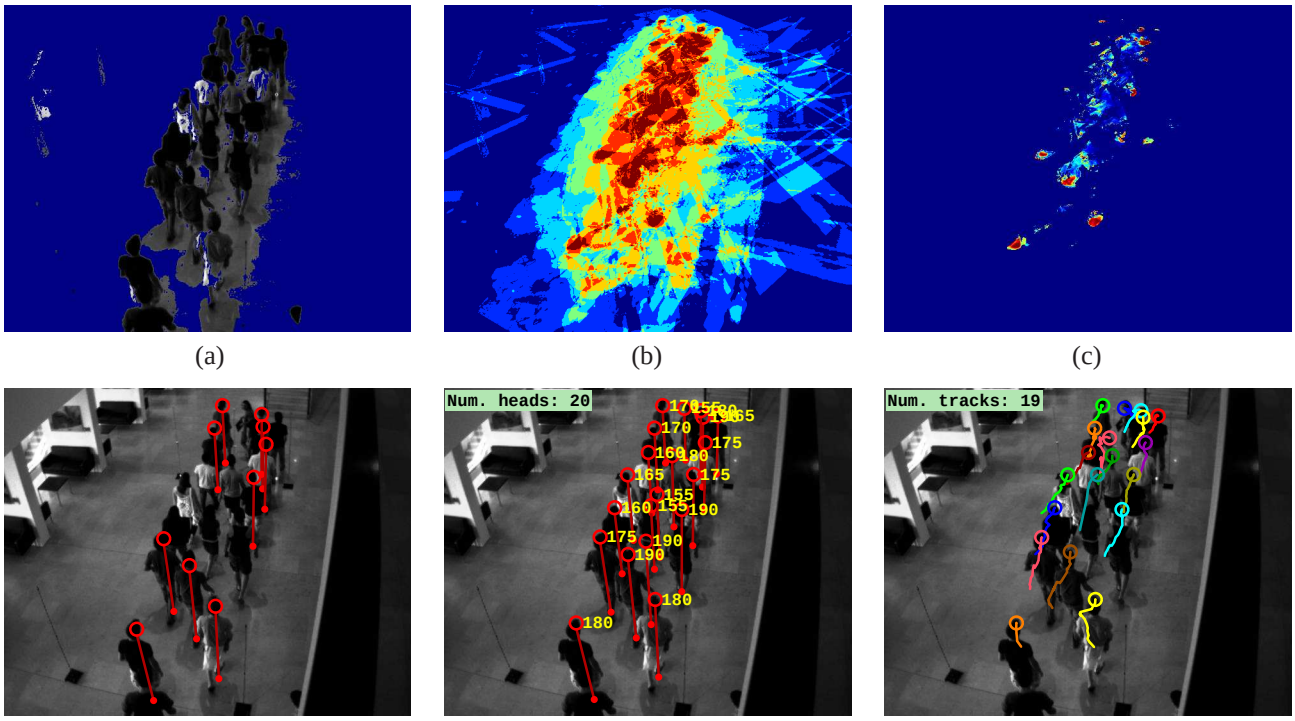


Figure 4. Intermediate results of head to detection. (a) Background subtraction on a single frame. (b) Aligned foreground of all views for a given height (color coded for the number of foregrounds in each hyper-pixel, where red is high). (c) Variance of the foreground hyper-pixels (red for low). (d) Detected head tops at a given height, and their projection to the floor . (e) The same as (d) for all heights. (f) Tracking results with 20 frame history.

3.2. Tracking

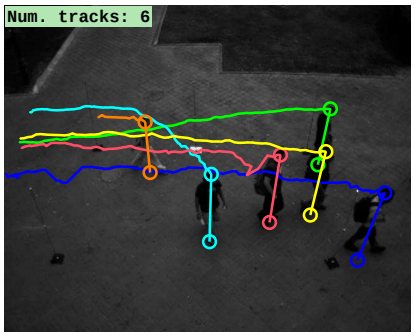
The input to the tracker for each time step consists of two lists of head top centers projected to the floor of the reference sequence. The difference between the two lists is the threshold used to compute them. Thus, the high threshold list will have less false positive head top detections but more false negative detections than the lower threshold list.

At the first stage of tracking, atomic tracks are computed using prediction of the feature location in the next frame based on its motion velocity and direction in previous ones. Tracking is performed using mainly the high threshold list. If several features are found within a small radius of the expected region, the nearest neighbor is chosen. If no feature is found within the region, the search is repeated using the lower threshold list. If this also fails, the process is repeated for the next time step, and only after two consecutive failures is the track abandoned. After all tracks have been matched to features in a given time step, the remaining unmatched features become candidates for new tracks. A new track is initialized only after tracking succeeds for at least two consecutive frames.

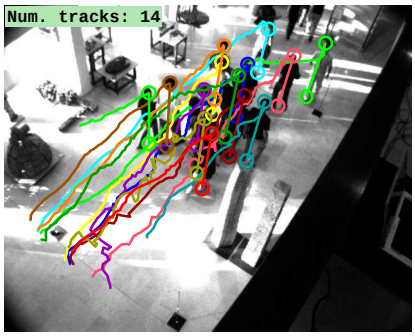
The result of the first stage of tracking is a large number of tracks, some of which are fragments of real trajectories and others which are false positives. The next stage combines fragments into long continuous tracks, leaving short unmatched tracks for deletion in the final stage.

Let tr_i and tr_j be two atomic tracks. The numbers of the first and last frames of a track are denoted by $f(tr_i)$ and $\ell(tr_i)$, respectively. The *overlap* of two tracks is defined as $overlap(tr_i, tr_j) = f(tr_j) - \ell(tr_i)$. Two tracks, tr_i and tr_j, are considered for merging if $-10 \leq overlap(tr_i, tr_j) \leq 40$. A merge score is computed for each pair of tracks that satisfies this condition. The score is a function of the following measures: m_1 – the amount of overlap between the tracks; m_2 – the difference between the two tracks' motion directions; m_3 – the direction change required by tr_i in order to reach the merge point with tr_j; m_4 – the height difference between tr_i and tr_j; m_5, m_6 – the minimal and average distances between corresponding points along the overlapping segments (or along the expected paths of the trajectories, in case of a negative overlap). The merge score is defined by: $score(tr_i, tr_j) = \frac{1}{6} \sum m_i / \widehat{m}_i$, where $\widehat{m}_i$ is the maximal expected value of the measure m_i.

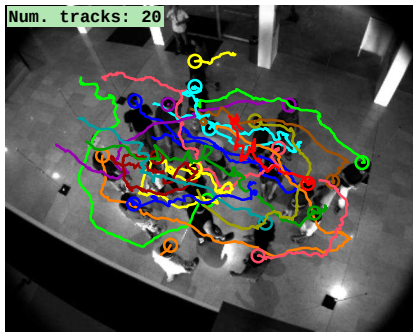
Finally, a consistency score is used to remove tracks that are suspected as false positives. This score is based on



S1



S3a



S5

Figure 5. Examples of tracked trajectories from three sequences. (For sequences **S1** and **S3a**, sticks connecting heads and their projections to the floor are displayed. For sequence **S5**, due to the complexity of the trajectories, only heads are displayed.)

weighted components which include the average change in speed, direction and height between any two consecutive time steps, and the track length. This heuristic successfully removes most of the phantom tracks. In addition, pairs of tracks that consistently move together, staying within a very small distance from each other are detected. In such cases, the shorter track, which is usually the shoulder, is deleted.

4. Experimental Results

To demonstrate the effectiveness of our method, we performed experiments on real video sequences under changing conditions. In Sec. 4.2 we describe the scenarios and the results of applying our method to several indoor and outdoor sequences with varying degrees of crowd density and challenging illumination conditions. In Sec. 4.3 we investigate how changing the number of cameras affects the tracking results.

4.1. Implementation and System Details

We used between 3 and 9 USB cameras (IDS uEye UI-1545LE-C), connected to 3 laptops. The cameras were placed around the scene, at distances of 2-3 meters from each other, with the vertical viewing angle of each camera rotated at 30^o relative to its neighbor. Horizontally, they were placed at an elevation of 6 meters, viewing the scene at a relatively sharp angle (45^o or more below the horizon). Detection and tracking were performed on an area of 3×6 meters. All test sequences were taken at a rate of 15 frames per second, with an image size of 640×512.

Cameras are calibrated using vertical poles placed at the corners of the scene, with blinking lights at the top, middle and bottom of each. Using the unique frequency of the lights on each pole, it is possible to generate planar homographies between the views for three planes parallel to the floor and to synchronize the sequences. These poles are also used to compute the floor projection homography matrices, B^h. In future work we intend to develop a calibration method that relies on tracked people in a non-dense environment, similar to [11].

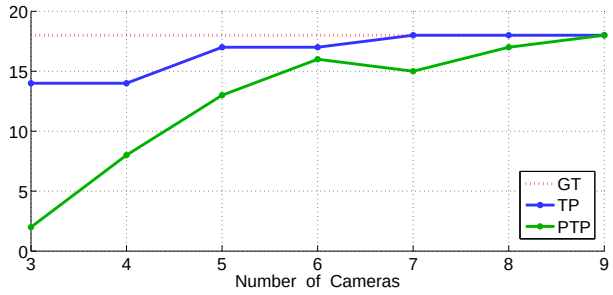
Seq	GT	TP	PTP	IDC	DR %	FN	FP
S1	27	26	23	3	98.7	1	6
S2	42	41	39	0	97.9	1	5
S3a	19	19	19	0	100.0	0	0
S3b	18	18	18	0	100.0	0	2
S3c	21	21	20	1	99.1	0	0
S4	23	23	22	0	99.1	0	1
S5	24	23	14	12	94.4	1	0
Total	174	155	16	16	98.4	3	14

Table 1. Tracking results on 7 Sequences (GT - Ground Truth; TP - True Positive, 75%-100% tracked; PTP - Perfect True Positive, 100% tracked, no ID changes along the trajectory; IDC - ID Changes; DR - Detection Rate; FN - False Negative; FP - False Positive).

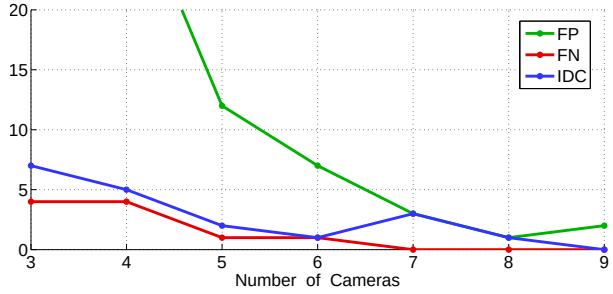
The algorithm was implemented in Matlab on gray level images. The algorithm's behavior is controlled by several parameters, all of which have a single global setting except for the hysteresis double thresholds. These are used to isolate high correlation (low variance) hyper-pixels of plane-aligned images, and are set manually for each sequence, since they depend on volatile factors such as the lighting conditions and the number of cameras.

4.2. Sequences and Results

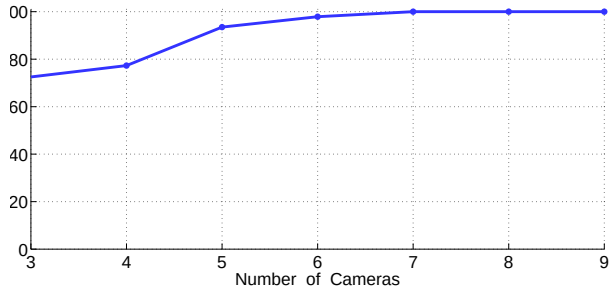
Below we describe the different scenarios used for testing our approach, and assess the system's performance on each of them. Frames from four of the sequences and a tail of the recovered trajectories are shown in Fig. 1 and in Fig. 5. Each detected person is marked by his head center and its projection to the floor. The tails mark the detected trajectories up to the displayed frame.



(a) Ground truth and true positives



(b) False positives, false negatives and ID changes



(c) Detection rate

Figure 6. System performance as a function of the number of cameras. Results improve as the number of cameras increases. When this number drops below 5, system performance deteriorates considerably.

The following evaluation criteria reflect both the success of recovering each of the trajectories and the success of assigning a single ID to each one. True Positive (*TP*): 75%-100% of the trajectory is tracked, possibly with some ID changes; Perfect True Positive (*PTP*): 100% of the trajectory is tracked, with a single ID (note that these trajectories are counted in TP as well); Detection Rate (*DR*): percent of frames tracked compared to ground truth trajectory, independent of ID change (and including false negatives); ID Changes (*IDC*): number of times a track changes its ID; False Negative (*FN*): less than 75% of the trajectory is tracked; False Positive (*FP*): a track with no real trajectory. Table 1 summarizes the tracking results.

S1: A long (1500 frames) relatively sparse (up to 6 concurrent trajectories) outdoor sequence using only 6 cameras which, due to physical limitations, are all collinear. The sequence was taken at twilight, and thus suffers from dim lighting and poor contrast. The tracking results are very good, except for a high false positive rate resulting from the low threshold chosen to cope with the low image contrast. Two of the three ID changes are caused by two people hugging each other, virtually becoming a single object for a while. Another person who enters and quickly leaves the scene is tracked only half-way, and counted as a false negative. Fig. 5**a** presents the tracking results on this sequence.

S2: A long (1100 frames) indoor sequence, with medium crowd density using 9 cameras. People in the scene move in groups (up to 9 people concurrently). Lighting conditions are very hard: bright lights coming in through the windows and reflected by the shiny floor create a highly contrasted background; long dark shadows interfere with foreground/background separation; inconsistent lighting within the scene significantly alters an object's appearance along different parts of its trajectory. In addition, tall statues are placed along the path, sometimes causing almost full occlusion. Despite these problems, the tracking quality is good, with only a single track lost, and most of the others perfectly tracked.

S3: Three excerpts from a longer sequence (200, 250 and 300 frames) with a very high crowd density, taken with 9 cameras. The scene is the same brightly lighted indoor scenario described in the previous sequence. The sequences contain 57 trajectories in total, with up to 19 concurrent. All of the people move very closely together in a single group and in the same direction (**S3a** & **S3b**), or split into two groups which closely pass each other in opposite directions (**S3c**). An additional difficulty is the inclusion of several bald-headed people in the sequence: the bright overhead lights falling on their heads gives them a different appearance in different views, resulting in a high hyper-pixel variance and a detection failure. Despite similar density, tracking results are significantly better than in sequence **S5**, partly because of the higher number of cameras, but mostly because of the more natural motion patterns displayed by the people. The detection rate is almost perfect (99.7%), and the error rate is very low (a total of 2 false positives, 0 false negatives and 2 ID changes for the three sequences combined). Fig. 5**b** presents the tracking results on sequence **S3a**.

S4: A high crowd density sequence (200 frames), taken using 6 cameras placed around the scene. Most of the people are visible at the same time (up to 19), and all of them move in the same direction, making separation based on motion impossible. Tracking results are very good: one of the tracks is detected late (30 frames after first appearing), while all the others are perfectly tracked.

S5: A very high crowd density sequence (200 frames) with complex motion taken with the same set-up as above. The

sequence begins with 21 people crowded into an $8m^2$ area, a density of over 2.5 people per m^2. People then start to move in an unnaturally complex manner - changing directions sharply and frequently, and passing very close to each other. The detection results are good, with a 94.4% detection rate and no false positives, but the tracking consistency is not as good, with almost half of the trajectories changing their ID at some point along their path. Fig. 5**c** presents the tracking results on this sequence. The tails demonstrate the complex motion of the people.

4.3. Varying the Number of Cameras

In theory, two or three cameras are sufficient for applying our method. In this experiment we test the effect of varying the number of cameras in one of our more challenging sequences, **S3b**. The results are summarized in Fig. 6. In general, both detection and tracking quality improve as the number of cameras increases. However, increasing this number beyond six has a negligible effect. The detection rate and the true positive detection remain high even when the number of cameras is decreased to three. As mentioned in Sec. 3 and demonstrated in Fig. 3**b**, decreasing the number of cameras may increase the number of accidental matchings, causing phantoms to appear. The effect of this phenomenon is apparent in Fig. 6**b**. The ambiguity caused by the presence of a large number of phantoms also affects other parameters, resulting in an increase in the number of ID changes and of false negative detections. We can therefore conclude that our tracker performs well when the number of cameras is sufficient for handling the crowd density. Otherwise, its performance gradually degrades as the number of cameras decreases.

5. Conclusions

We suggest a method based on a multiple camera system for tracking people in a dense crowd. Our main contribution is the use of multiple height homographies for head top detection, which makes our method robust to severe and persistent occlusions, and to challenging lighting conditions. Most of the false positives generated by this method are removed by a heuristic tracking scheme.

In the future we intend to investigate automatic setting of system parameters and to consider a distributed implementation of our algorithm. Another promising direction is to combine our algorithm with human body segmentation methods, to assist in false positive removal.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999. 2

[2] Y. Bar-Shalom and A. G. Jaffer. *Tracking and Data Association*. Academic Press, San Diego, Calif, USA,, 1998. 3

[3] Q. Cai and J.K. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *PAMI*, 21(11):1241–1247, 1999. 2

[4] O.D. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Boston MA, 1993. 1

[5] S. Fleck, Buschm F., P. Biber, and W. Straber. 3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d. In *CVPR-06 workshop*, page 118, 2006. 2

[6] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. 2

[7] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In *ICCV-2001*, pages 34–41, 2001. 2, 3

[8] S.M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV-06*, pages IV: 133–146, 2006. 2

[9] S.M. Khan, P. Yan, and M. Shah. A homographic framework for the fusion of multi-view silhouettes. In *ICCV-2007*, 2007. 3

[10] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *International Workshop on Visual Surveillance*, 2000. 2

[11] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *PAMI*, 22(8):758–767, August 2000. 6

[12] B. Leibe, Seemann. E., and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR-2005*, volume 1, pages 878–885, 2005. 2

[13] A. Micilotta, E. Ong, and R. Bowden. Detection and tracking of humans by probabilistic body part assembly. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2005. 1

[14] A. Mittal and L. Davis. Unified multi-camera detection and tracking using region matching, 2001. 1, 2

[15] W. Qu, D. Schonfeld, and M. Mohamed. Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. *EURASIP J. Appl. Signal Process.*, 2007(1):21–21, 2007. 3

[16] M. Quaritsch, M. Kreuzthaler, B. Rinner, Bischof. H., and B. Strobl. Autonomous multicamera tracking on embedded smart cameras. *Journal on Embedded Systems*, 2007. 2

[17] V. Rabaud and S. Belongie. Counting crowded moving objects. *CVPR-06*, 1:705–711, 2006. 2

[18] M. D. Rodriguez and M. Shah. Detecting and segmenting humans in crowded scenes. In *MULTIMEDIA-07*, pages 353–356, 2007. 1

[19] K Smith, D. Gatica-Perez, and J. Odobez. Using particles to track varying numbers of interacting people. In *CVPR-2005*, pages 962–969, 2005. 2

[20] E. Trucco and K. Plakas. Video tracking: A concise survey. *Oceanic Engineering, IEEE Journal of*, 31(2):520–529, 2006. 3

[21] P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005. 1

[22] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007. 2

[23] A. Yilmaz, O. Javed, and S. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006. 3

[24] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *CVPR-2007*, 2007. 3

[25] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *PAMI*, 26(9):1208–1221, 2004. 1, 2