Int J Comput Vis (2010) 88: 129–143
DOI 10.1007/s11263-009-0307-0

Tracking in a Dense Crowd Using Multiple Cameras

Ran Eshel · Yael Moses

Received: 8 February 2009 / Accepted: 3 November 2009 / Published online: 17 November 2009

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 the people in a densely crowded scene 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.

Keywords Tracking · Detection · Multiple-view

1 Introduction

People tracking is a well-studied problem in computer vision, mainly, but not exclusively, for surveillance applications. One of the main challenges encountered by tracking methods is the severe and persistent occlusion prevalent in images of a dense crowd (as shown in Fig. 1). Most existing tracking methods use a single camera, and thus do not cope well with crowded scenes. For example, trackers based on a human shape model such as Rodriguez and Shah (2007) or Zhao and Nevatia (2004) will encounter difficulties since body parts are not isolated, and may be significantly occluded. Multiple camera tracking methods often perform segmentation in each view separately, and are thus susceptible to the same problems (e.g., Mittal and Davis 2001 or Krumm et al. 2000).

Fig. 1 Tracking results using combined data from all views, overlaid on the reference frame (original frame can be seen in Fig. 2)

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. Our method avoids occlusion by only tracking heads. We place a set of cameras at a high elevation, from which the heads are al-

R. Eshel (✉) · Y. Moses
Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya 46150, Israel
e-mail: eshel.ran@idc.ac.il

Y. Moses
e-mail: yael@idc.ac.il

Fig. 2 Six views of a single scene taken at the same time

most always visible. Even under these conditions, head segmentation using a single image is challenging, since in a dense crowd, people are often merged into large foreground blobs (see Fig. 7a). To overcome this problem, our method combines information from a set of static, synchronized and partially calibrated cameras, with overlapping fields of view (see examples in Fig. 2).

We rely on the fact that the head is the highest region of the body. A head top is roughly 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 are transformed using a planar homography (Faugeras 1993) to align the projection of the plane at that height in all images. (Note that each of the foreground regions may contain several people.) Intensity correlation in the set of transformed frames is used to detect the candidate blobs. In Fig. 4 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 finding the centers of these blobs, and projecting them to the floor, multiple detections of the same person at different heights can be removed. At the end of this phase we obtain, for each time step, 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 tracks and filter them out. Tracking results can be seen in Fig. 1.

The main contributions of this paper are: (1) The use of multiple height homographies for head top detection; (2) The fusion of information from multiple views through intensity correlation. The described method 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. The method was tested on indoor and outdoor sequences with challenging lighting conditions, and was successful in tracking up to 21 people walking in a small area (2.5 people per m^2). A preliminary version of this paper appeared in Eshel and Moses (2008).

The rest of the paper is organized as follows: in the next section we present a review of previous work. Section 3 describes the two main elements of our method: the head top detection phase, and the tracking phase. Experimental results for real-world video sequences are presented in Sect. 4.

Finally, in Sect. 5, we discuss these results, and suggest future research directions.

2 Related Work

There is extensive literature on multiple target tracking, and specifically on people tracking, mostly from a single view. In Sect. 2.1 we review some single camera detection and tracking methods, and discuss their limitations when applied to densely crowded scenes. The use of multiple cameras for people tracking is becoming more common, and such methods are described in Sect. 2.2. Finally, in Sect. 2.3, we give an overview of homography based methods, which are most similar to ours.

2.1 Single Camera Approaches

Until recent years, the bulk of research in the field of people detection and tracking concentrated on using a single camera to track a small number of subjects, most commonly detected using machine learning techniques. Earlier methods try to match the image against templates of full human figures (Felzenszwalb 2001; Gavrila and Philomin 1999; Papageorgiou and Poggio 1998), and therefore do not perform well when subjects are even partially occluded. More recent methods detect body parts separately, by breaking down regions of interest into sub-regions that correspond to local features (Shashua et al. 2004), by using a full-body representation but allowing interpolation between local parts seen on different training objects (Leibe et al. 2005), or by boosting weak classifiers based on edgelet features and combining them to form a joint likelihood model (Wu and Nevatia 2007). While these local feature based methods are less sensitive to occlusions, they still require that most of the tracked person will be visible most of the time.

Another class of single camera detection and tracking algorithms rely on motion information rather than on appearance, by looking for repetitive motion (Polana and Nelson 1994) or by tracking simple image features and probabilistically grouping them into clusters representing independently moving entities (Brostow and Cipolla 2006). Much like the full body detection methods discussed above, these methods rely on an almost full visibility of the object, and are thus intolerant to occlusions. Viola et al. (2005) improve on previous results by integrating motion information with appearance information. While using a combination of motion and appearance results in a more robust approach, it is still limited when a dense crowd is considered. Under difficult conditions, which preclude the use of either motion or appearance, their combination cannot be expected to produce significantly better results.

Several approaches employ a Bayesian framework to improve tracking, relying on relatively simple detection methods. Some use various types of particle filters (e.g. Isard and MacCormick 2001; Smith et al. 2005), while others use Markov Chain Monte Carlo approaches to sample the solution space efficiently (Yu et al. 2007). These, like other single camera methods, are inadequate for handling highly dense crowds such as those considered in this paper, due to severe occlusion which results in large foreground regions comprised of multiple people. For example, a suggested comparison between our method and the state-of-the-art single view tracking system developed by Wu et al. could not be performed, since their method was reported to be inapplicable under these challenging density and illumination conditions.[1]

Recently, Ali and Shah (2008) suggested applying methodologies from the field of evacuation dynamics for tracking people in highly dense crowds. By relying on the static structure of the scene and on the dynamic behavior of the crowd in the vicinity of the tracked person, a strong prior on the person's motion is assumed. This produces impressive tracking results for people that move along with the crowd. However, any deviation from the expected behavior (e.g. a person moving in the opposite direction from the crowd), will most likely result in a detection failure, since in that case the prior will preclude detection, rather than facilitate it.

2.2 Multiple Camera Approaches

Due to the inherent limitations of single camera trackers when applied to dense crowds, or to environments where no single camera position provides an unobstructed view of the scene, new approaches attempt to fuse the data from multiple cameras. Traditionally, multiple cameras were used for extending the limited viewing area of a single camera. In this case, tracking is performed separately for each camera, and the responsibility of tracking a given subject is transfered from one camera to another (Cai and Aggarwal 1999; Kettnaker and Zabih 1999; Quaritsch et al. 2007). This approach does not offer any improvement in tracking results, since at any given time, only a single camera is responsible for tracking. To mitigate the effects of occlusion, some methods use multiple cameras with overlapping fields of view. Kobayashi et al. (2006) and Nummiaro et al. (2003) use multiple cameras to robustly track a single target, but most multiple camera methods attempt to negotiate more challenging scenarios.

Krumm et al. (2000) use pairs of cameras to resolve ambiguity using 3D stereo information. Their method is based on background subtraction, and is hence limited when a

[1] Personal communication.

(a) (b) (c)

Fig. 3 Intensity correlation improves detection. (a) Combined foreground from all views, without using intensity correlation, at ground plane. Detection result is a single large blob, containing most of the people in the scene. (b) Same as (a), but for height 170 cm. Detection results at the head height are significantly improved compared to those at the ground plane, but still the number of false positives, specifically in the dense areas, is too large to facilitate proper separation of the different people. (c) Our method: detection using intensity correlation at height 170 cm. All people at the given height are detected, and there are no false positives. (Note that in order to detect all of the people in the scene, detection results from multiple heights must be combined, as described in Sect. 3.1.2)

dense crowd is considered. Orwell et al. (1999) use color histograms to maintain consistent labeling of tracked objects. While this may provide reliable cues for coordinating between different cameras tracking the same object, the color histograms will be significantly altered when objects are occluded, and therefore this approach cannot be used for tracking in a dense crowd. Mittal and Davis (2001) employ a higher level of collaboration between cameras, by matching foreground blobs from different views along epipolar lines. Initial separation of the foreground into regions is performed using a simple color segmentation algorithm. The main limitation of their method is its reliance on the assumption that different people within a single foreground blob are separable based on color segmentation alone. This assumption does not always hold, since people often wear similarly colored clothes. The same authors later introduced a much more discriminative color model (Mittal and Davis 2003), which provides better distinction between different people, but this will still fail in tracking people with very similar appearance, such as sports fans wearing team jerseys.

Du and Piater (2007) track targets in each camera separately using particle filters, and then pass the results to combined particle filters on the ground plane. Additionally, tracking results from the ground plane are passed back to each camera, to be used as boosted proposal functions. To alleviate the need for precise foot positioning, target location on the ground plane is found by intersecting the targets' principal axes. The main limitation of this method is the dependence on separate trackers in each camera, which are limited in their ability to handle occlusion. Intersection of principal axes is also used by Kim and Davis (2006). They perform segmentation of foreground regions using a viewpoint-independent appearance model, and iteratively combine segmentation results with the axes intersection point. This detection process is embedded within a particle filter framework for tracking. Fleuret et al. (2007) use a generative model which represents people as rectangles to approximate the probabilities of occupancy at every location on the ground plane. These probabilities are combined using a greedy algorithm which tracks each target over a long period of time, and uses a heuristic approach to avoid switching labels between targets. However, since the initial occupancy map is generated based on the results of a background subtraction algorithm, the perfect tracking results achieved in their experiments will diminish significantly in high crowd densities.

2.3 Homography-Based Multiple Camera Approaches

The use of multiple plane homographies for detection, which is a fundamental part of our method, was previously suggested by Garibotto and Cibei (2005). Since their method attempts to completely reconstruct the objects, but includes no mechanism for handling occlusions, its utilization is only feasible for sparse scenes.

The method most similar to ours for detecting people from multiple cameras was proposed by Khan and Shah (2006), except that it detects people's feet, rather than their heads. They align the foreground of the ground plane in images taken from a set of cameras with overlapping fields of view, to detect people's feet. Their method handles occlusions by applying the homography constraint, which states that any 3D point lying inside the foreground object in the scene will be projected to a foreground pixel in every view. This works quite well for moderately crowded scenes, but seems inadequate for handling higher crowd densities: On one hand, tracking people's feet rather than their heads precludes the use of intensity value correlation, since the

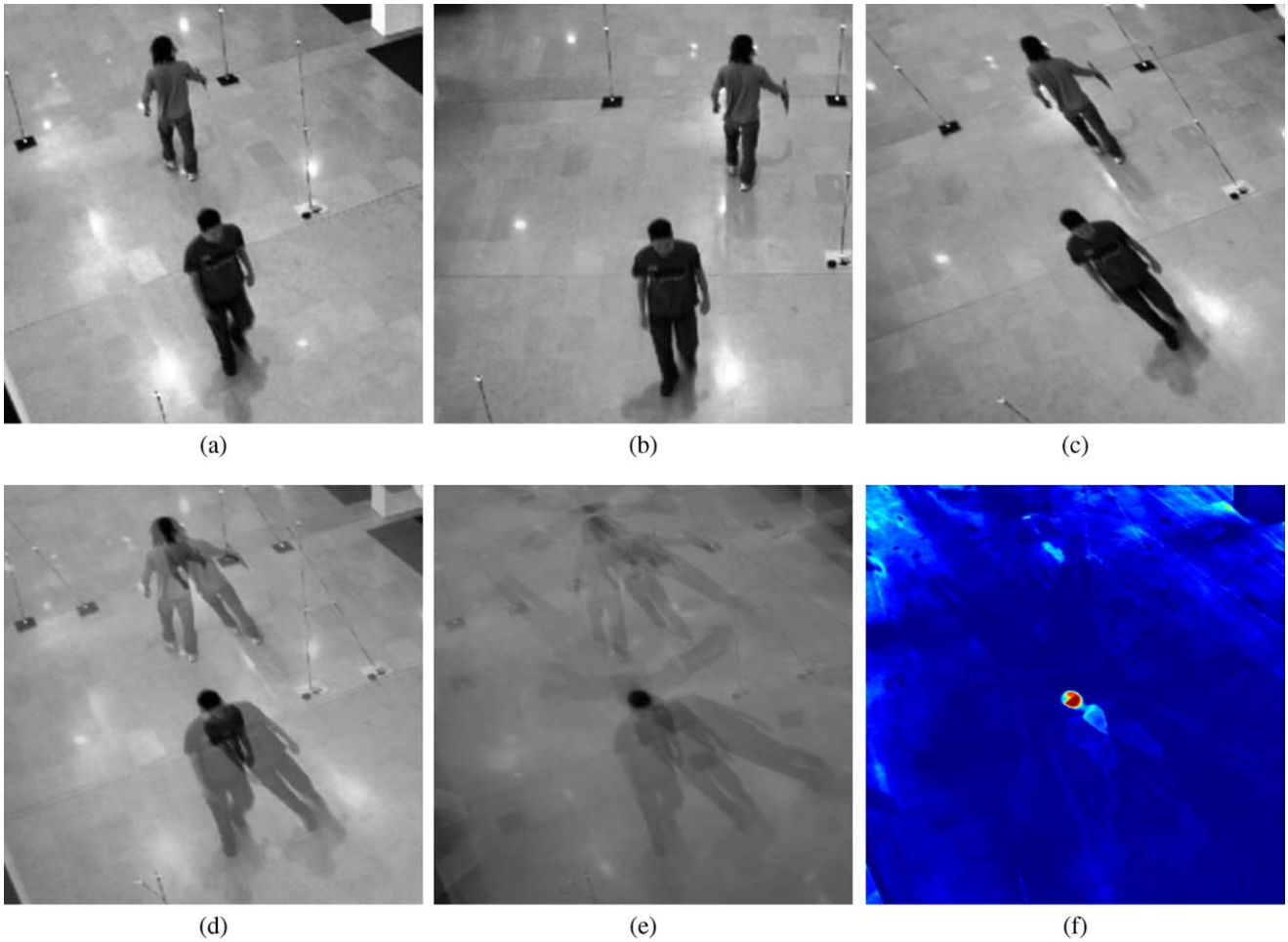


Fig. 4 2D patch detection demonstrated, for clarity, on a single, unoccluded person. (The second person, at *the back of the image*, is 8 cm shorter, therefore the top of his head does not lie on the plain, and is thus not detected at this height.) (a, b) Two views of a single scene taken at the same time. (c) Applying homography transformation to image (b) to align points on the 3D plane at the head-top height with their counterparts in image (a). (d) Image (c) overlaid on image (a). (e) Overlay of additional transformed images. (f) Variance map of the hyper-pixels of image (e), color coded such that *red* corresponds to a low variance

occlusion of the feet in a dense crowd is likely to cause many false negative detections. On the other hand, detection based solely on foreground/background separation of images rather than on a more discriminative correlation of intensity values can result in false positive detections (as explained in Sect. 3.1.4, and demonstrated in Fig. 3).

Recently, Arsic et al. (2008) suggested applying the same concept to planes at multiple heights. Indeed, since they also rely on detection of foreground regions, their method suffers from a high false positive rate, which is made worse by the cumulation of results from multiple heights, specifically lower heights where occlusion is more severe.

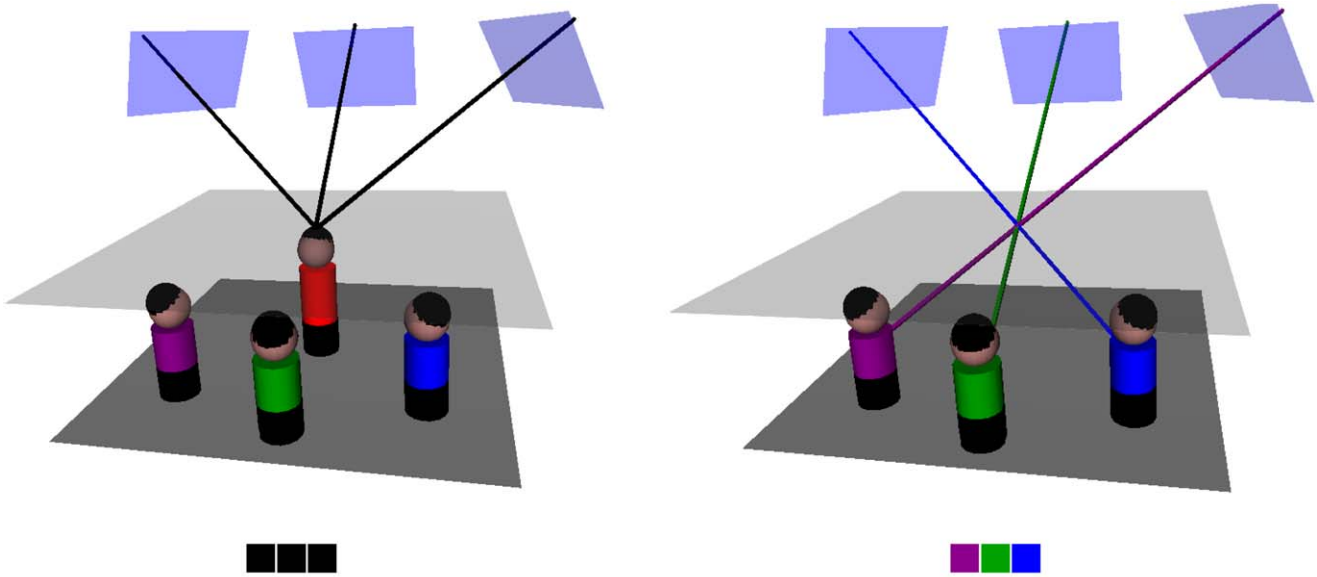
Khan et al. (2007) use multiple height homographies for 3D shape recovery of non-occluded objects. Several other methods have utilized multiple cameras viewing a single object from different directions for 3D reconstruction, based on the visual hull concept (Laurentini 1994), or on constructing a space occupancy grid (Cheung et al. 2000, Franco and Boyer 2005). However, none of these methods was used for tracking, or in the presence of occlusion.

For a more thorough discussion of tracking techniques, we refer the reader to the comprehensive survey by Yilmaz et al. (2006).

3 The Method

We assume a set of synchronized and partially calibrated cameras overlooking a single scene, where head tops are visible. The setup, described in Sect. 4.1, allows to compute homographies between views and between different heights within the same view.

Initially, head top centers and their heights are detected (each represented by a single feature point), and projected



(a) Low variance - true positive

(b) High variance - true negative

Fig. 5 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) No person to detect: points belonging to different objects have different colors. This results in high hyper-pixel intensity variance, which prevents false positive detection

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 a 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. Figure 7 shows intermediate results of the method described below, and Algorithm 3.1 gives a synopsis of the algorithm.

3.1.1 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. 4 for a single person at a given height.

Consider n synchronized cameras. Let S_i be the sequence taken by camera i, with S_1 serving as the reference sequence. Let π^h be a plane in the 3D scene parallel to the 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 correspondences given by the partial calibration, the homography matrices $A^h_{i,1}$ can be computed for any height h (see Appendix).

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 us to compute $\bar{q}^h$:

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

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 plane π^h (see Fig. 5a). A low correlation can be expected for other points provided that the scene is not homogeneous in color (see Fig. 5b). 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

Algorithm 3.1 2D patch detection at time t

foreach image $S_i(t)$ **do**
  Detect foreground pixels using background subtraction
end for
// Detect head top centers at each height separately:
for $h \in H$ **do**
  // Create hyper-pixel intensity map $\bar{S}(t)^h$ for height h at time t:
  foreach point $q \in S_1(t)$ **do**
    for $i = 1$ to n **do** // Create hyper-pixel $\bar{q}^h$
      $\bar{q}^h(i) \leftarrow I_i(A^h_{1,i} q)$
    end for
    Compute intensity variance of $\bar{q}^h$
  end for
  Perform hysteresis thresholding on $\bar{S}(t)^h$
  Perform segmentation on $\bar{S}(t)^h$ to create 2D patches
  // Find head top centers, and project to ground plane:
  foreach patch $\bar{p}_j \in \bar{S}(t)^h$ **do**
    $\bar{c}_j \leftarrow$ center of $\bar{p}_j$
    $c_j \leftarrow$ projection of $\bar{c}_j$ to ground plane
    $C(t)^h \leftarrow C(t)^h \cup \{c_j\}$ // Add to head centers list for height h
  end for
end for
// Find the highest 2D patch at each floor location:
for $h \in H$ **do** // Traverse heights from top to bottom
  foreach projected patch center $c_j \in C(t)^h$ **do**
    $L(t) \leftarrow L(t) \cup \{c_j\}$ // Add to final list of heads
    Delete all $c_i \in C(t)^{h'}$ s. t. $h' < h$ and $\|c_i - c_j\| \leq thresh$
  end for
end for
return $L(t)$

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 (Lloyd 1982). The number of clusters is determined by dividing the blob size by the expected head size. The centers of the 2D patches are then used for further processing.

A possible source of false positive detections is homogeneous background. For example, in an outdoor scene, the texture or color of the ground may be uniform, as may be the floor or walls in an indoor scene. We therefore align only the foreground regions, computed using a simple background subtraction algorithm (which subtracts each frame from a single background frame, taken when the scene was empty).

3.1.2 Finding the Highest 2D Patch

The process of detecting 2D patches is repeated for a set $H = \{h_1, \ldots, h_n\}$ of expected people heights. The set is taken at a resolution of 5 cm, within the range 150–190 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 or more of the considered heights. However, a single person might be detected as patches at several heights, and all but the highest one should be removed. To do so, we compute the foot location of each of the 2D patches as would appear in the reference sequence.

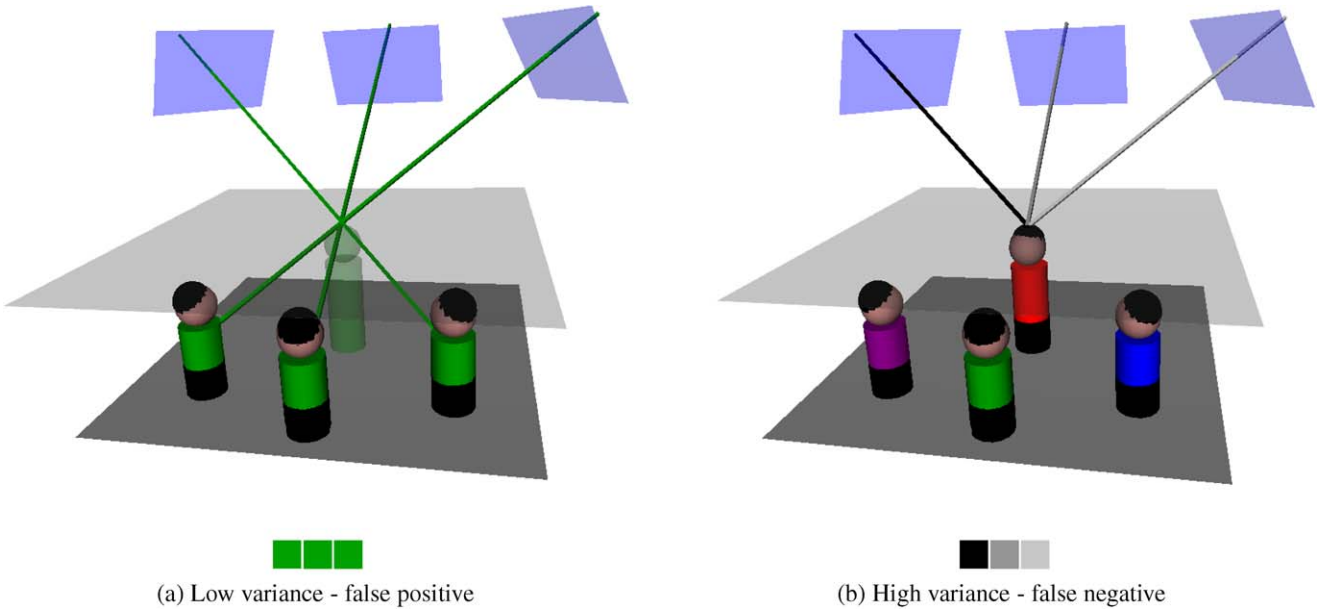
The foot location is assumed to be the orthogonal projection of a 2D patch at a given height h to the floor. The projection is computed using a homography transformation from the reference sequence to itself. The homography aligns the location of each point on the plane π^h in the reference image with 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 based on the partial calibration assumption of our system. For a 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.

3.1.3 Applying Multiple Thresholds

The results of the head top detection process described above depend on the choice of threshold to be applied to the hyper-pixel intensity variance. To reduce this dependence, the complete process is performed twice, with two different thresholds. First, a relatively high threshold is applied: this produces reliable results, with few false positive detections, but possibly some false negatives. Then, the process is repeated with a lower threshold to recover the patches that were missed in the first stage, resulting in an increase in false positive detections. The results from both stages are sent to the tracker, where tracks are formed from the high threshold results, and the low threshold results are used to fill in gaps within the tracks.

3.1.4 Expected Problems

'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, and accidental intensity correlation of aligned frames may be detected as head



(a) Low variance - false positive

(b) High variance - false negative

Fig. 6 Possible causes of misdetection. (a) 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. (b) When the surface of the object is reflective (for example, if the person is bald), the same point on the object may seem to have different colors when viewed from different viewpoints, resulting in high hyper-pixel intensity variance, and a false negative detection. Note that this figure is identical to Fig. 5a except for the ray colors, which differ due to specularity

tops. Figure 6a illustrates how plane alignment can correlate non-corresponding pixels originating from different people who happen to be wearing similarly colored clothes. In this case, rays intersect in front of the people, and the created phantom is taller. Similarly, shorter phantoms may appear if the rays intersect behind the people. Note that if only background/foreground values are used, as in Khan and Shah (2006), such accidental detections will occur even if people are wearing different colors (as in Fig. 5b). Our method will not detect a phantom in this case, since it uses intensity value correlation.

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, as demonstrated by the experimental results (see Sect. 4.3).

Phantoms are removed in the tracking phase, by filtering out tracks that exhibit abnormal motion behavior. Phantom removal can be further improved by utilizing human shape detection methods, but this is beyond the scope of this paper.

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. Each list is computed using a different threshold. 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 the high threshold list. If several features are found within a small radius of the predicted location, the nearest neighbor is chosen. If no feature is found within this region, the search is repeated using the lower threshold list. Failure to find the feature in either list is considered a negative detection. The termination of tracks is determined by the number of successive negative detections. After all tracks have been matched to features in a given time step, the remaining unmatched features are considered as candidates for new tracks. Tracks are initialized from these candidates only after two or more consecutive positive detections.

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 time stamps of the first and last frames of a track are denoted by $f(tr_i)$ and $\ell(tr_i)$, respectively. The time overlap of two tracks is defined as:

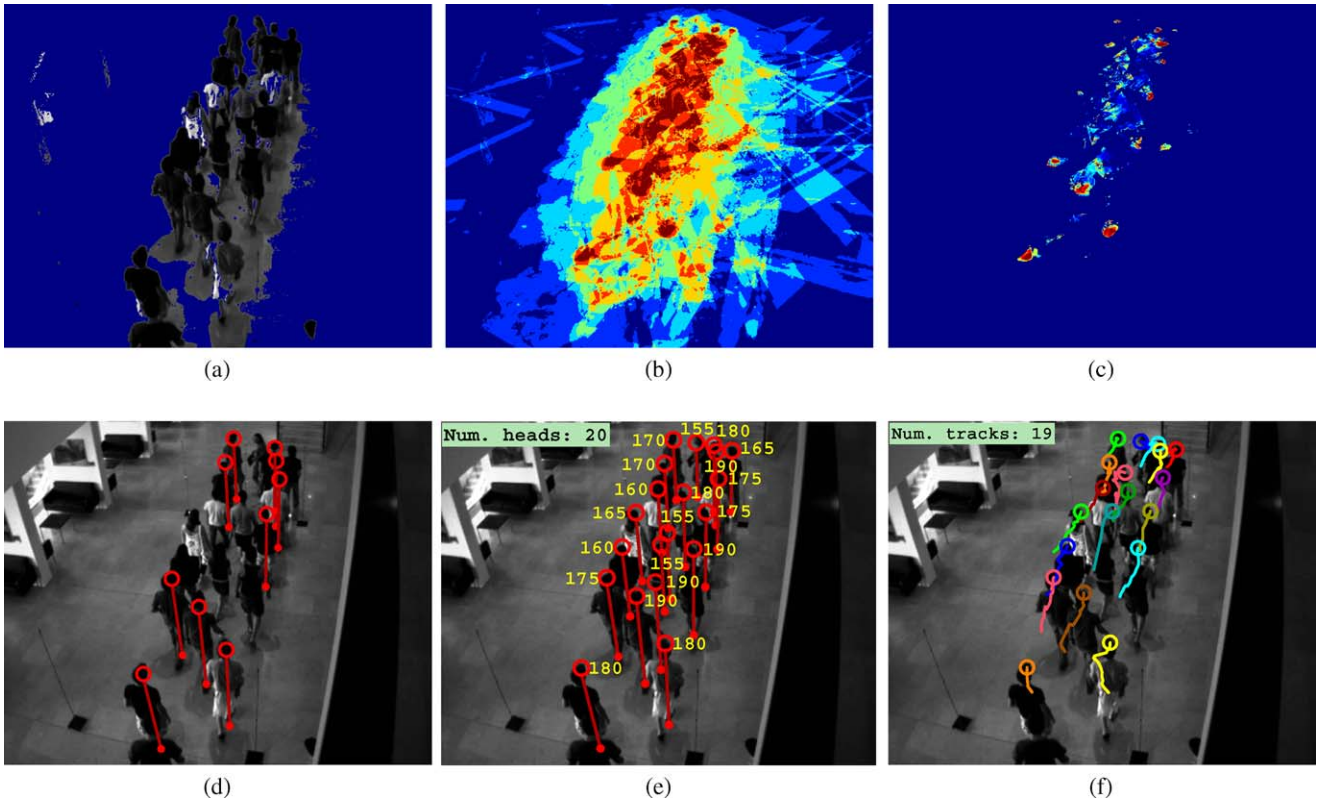


Fig. 7 Intermediate results of head top detection. (a) Background subtraction on a single frame. (b) Aligned foreground of all views for a given height (color coded for the number of foregrounds in each hyper-pixel, where *red* is high). (c) Variance of the foreground hyper-pixels (*red* for low). (d) 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

$$\text{overlap}(tr_i, tr_j) = \min(\ell(tr_i), \ell(tr_j)) - \max(f(tr_i), f(tr_j)) + 1$$

If instead of an overlap, there exists a gap between the two tracks, then the value of *overlap* will be negative. Two tracks, tr_i and tr_j, are considered for merging if:

$$-10 \leq \text{overlap}(tr_i, tr_j) \leq 40$$

A *match likelihood score* is computed for each pair of tracks that satisfies this condition. A high value indicates a low probability that the two tracks belong to the same real trajectory. The score is a normalized sum of the following measures (see Fig. 8):

- m_1—The number of overlapping frames between the two tracks, or the size of the gap between them (in the case of a negative overlap). The value is 0 for a zero overlap, and increases as the overlap or the gap between the tracks grows. A gap incurs a higher penalty than an overlap with the same size (see Fig. 8c).
- m_2—The difference between the two tracks' motion directions. A low value, indicating that the tracks are moving in roughly the same direction, increases the probability that they belong to the same real trajectory (see Fig. 8a).
- m_3—The direction change required by tr_i in order to reach the merge point with tr_j. Even if the locations and motion directions of the two tracks are similar, joining them might require a sharp change in direction. Tracks belonging to the same trajectory are expected to require a small direction change (see Fig. 8b).
- m_4—The height difference between tr_i and tr_j. The heights compared are the average heights of the two tracks, which are assumed to be very similar if both tracks follow the same person.
- m_5—The minimal distance between corresponding points along the overlapping segments (or along the expected paths of the trajectories, in case of a negative overlap). The distance between the tracks is highly indicative of whether they should be merged or not (see Fig. 8c).
- m_6—The average distance between corresponding points along the overlapping segments. Since the minimal distance is relatively volatile, and might be influenced by outliers, the average distance, which is more robust, is also considered (see Fig. 8c).

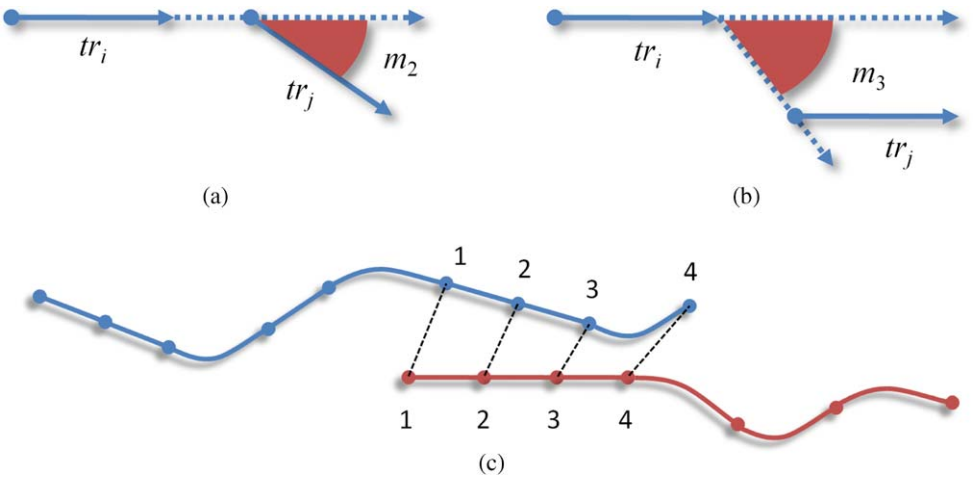


Fig. 8 The measures used to determine the likelihood that two tracks belong to the same real trajectory. (a) m_2—the difference between the two tracks' motion directions. (b) m_3—the direction change required by tr_i in order to reach the merge point with tr_j. (c) m_1—the number of overlapping frames between the tracks (in this example, 4); m_5—the minimal distance between corresponding points (in this example, the points designated 3 in each track); m_6—the average distance between corresponding points along the overlapping segments

The match likelihood score is defined by:

$$\text{score}(tr_i, tr_j) = \frac{1}{6}\sum m_i/\widehat{m}_i$$

where $\widehat{m}_i$ is the maximal expected value of the measure m_i. The goal of this normalization is to adjust the six different measures to a comparable scale.

Finally, tracks suspected as false positives are removed, based either on their length (very short tracks are assumed to be false positives), or on their motion consistency score. To compute this score, the change in speed, direction and height between any two consecutive time steps is computed, and averaged over the entire track length. These three measures are then normalized and summed into a single score, in a manner similar to the computation of the match likelihood score above. This heuristic successfully removes most of the phantom tracks. In addition, pairs of tracks that consistently move together, staying within a very small distance of each other, are assumed to belong to the same person (e.g. separate detections of the head and of the shoulder), and one of them is deleted.

To summarize, we handle false negative detections of partial trajectories by allowing a small number of missed detections when computing atomic tracks, and then combining atomic tracks into longer tracks. In both cases we use common assumptions on motion speed and direction to resolve ambiguities. False positive detections are removed using heuristics based on length and on motion consistency.

4 Experimental Results

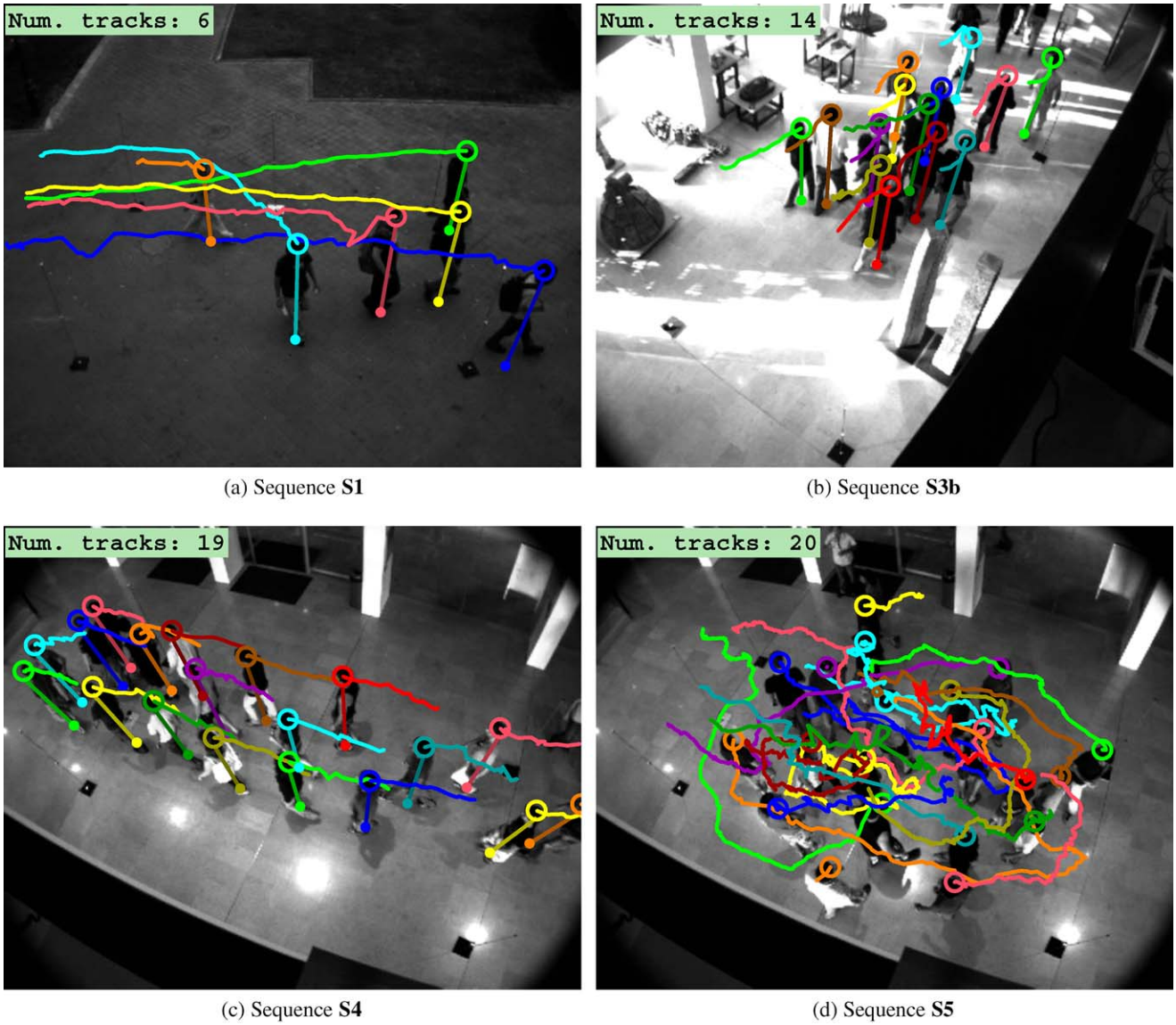
To demonstrate the effectiveness of our method, we performed experiments on real video sequences under changing conditions. In Sect. 4.2 we describe the scenarios and the results of applying our method to several indoor and outdoor sequences with varying degrees of crowd density and challenging illumination conditions. In Sect. 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 Intel Core Duo 1.7 MHz laptops. The cameras were placed around the scene, 2–3 meters apart, with the vertical viewing angle of each camera rotated at 30° relative to its neighbor. Horizontally, they were placed at an elevation of 6 m, viewing the scene at a relatively sharp angle (45° or more below the horizon). Detection and tracking were performed on an area of 3 m × 6 m. All test sequences were taken at a rate of 15 frames per second, with an image size of 640 × 512.

The cameras were calibrated using a novel method described in Goldschmidt and Moses (2008). Vertical poles are placed at the corners of the scene, with blinking LEDs at the top, middle, and bottom of each. The LEDs on each pole blink at a unique frequency, which can be detected and used for generating correspondences between all views. From these correspondences, it is possible to extract planar homographies between the views for planes parallel to the ground at any height (see Appendix). The same data is also used to synchronize the sequences, and to compute the ground plane projection homography matrices, B^h.

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 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 number of cameras used, their relative positions, the lighting conditions in the scene, and the accuracy of the homographies.



(a) Sequence **S1**

(b) Sequence **S3b**

(c) Sequence **S4**

(d) Sequence **S5**

Fig. 9 Examples of tracked trajectories from four sequences. (a) *Circles* mark the tracked heads, and *straight lines* connect the heads to the feet. *The tails* represent the full tracking history of each person. (b, c) In a denser crowd, only a 20 frame history is displayed for each person. (d) To show the complexity of the motion paths, only heads are displayed, but with full tracking history

4.2 Sequences and Results

Below we describe the different scenarios used for testing our approach, and assess the system's performance.

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 changes (false negative tracks are also included, counted as having 0 tracked frames)
- 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. Examples can be seen in Fig. 1 and in Fig. 9, where each detected person is marked by his head center, and its projection to the

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)

Sequence	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	171	155	16	98.4	3	14

ground plane. The tails mark the detected trajectories up to the displayed frame.

We next describe each sequence in detail[2]:

S1: A 1500 frame long, 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. Figure 9a presents the tracking results on this sequence.

S2: A 1100 frame long indoor sequence, with medium crowd density using 9 cameras. The scene contains up to 9 people concurrently, some of them moving together in groups. 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 (200, 250 and 300 frames long) from an indoor sequence with a 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** and **S3b**), or split into two groups which pass close to 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 a bald head give it a different appearance in different views, resulting in a high hyper-pixel variance and a detection failure. Tracking results are good: 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). Figure 9b presents the tracking results on sequence **S3b**. Figures 1 and 12 present the tracking results on sequence **S3c**.

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, yielding a 99.1% detection rate. There are no false negatives and no ID changes, and only a single false positive. Figure 9c presents the tracking results on this sequence.

S5: A high crowd density sequence (200 frames) with complex motion taken with the same setup as above. The sequence begins with 21 people crowded into an 8 m^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. Figure 9d presents the tracking results on this sequence. The tails demonstrate the complex motion of the people.

[2]Tracking results can be seen in: ftp://ftp.idc.ac.il/Pub/Users/CS/Yael/CVPR-2008/CVPR-2008-results.zip.

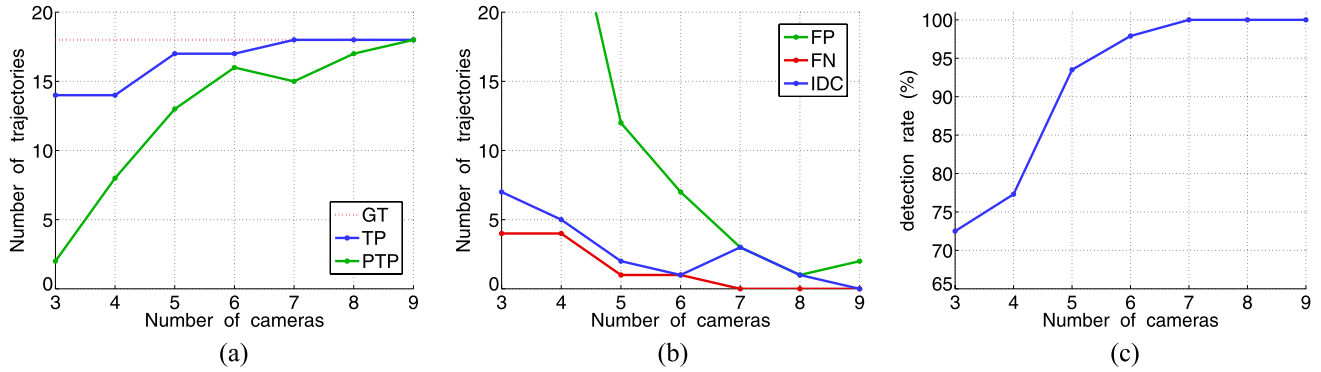


Fig. 10 System performance as a function of the number of cameras. Results improve as the number of cameras increases. When this number drops below 5, system performance deteriorates considerably. (a) Ground truth and true positives. (b) False positives, false negatives and ID changes. (c) Detection rate

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. 10. 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 Sect. 3.1 and demonstrated in Fig. 6a, 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. 10b. 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 Conclusion

We suggest a method based on a multiple camera system for tracking people in a dense crowd. The use of multiple cameras with overlapping fields of view enables robust tracking of people in highly crowded scenes. This may overshadow budget limitations when essential or sensitive areas are considered. The sharp decline in camera prices in recent years may further increase the feasibility of this setup.

Our main contributions are the use of multiple height homographies for head top detection, and the fusion of information from multiple views through intensity correlation, which make 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.

Possible directions for future work include augmenting our detector with a human shape detector to reduce the number of false positives, and using appearance features to improve track consistency and thus reduce the number of ID changes.

Acknowledgements This research was supported by the Israel Science Foundation (grant No. 1339/05). We would like to thank Ran Goldschmidt for assisting in data capture and in calibration and synchronization of the sequences.

Appendix: Transforming Between Views

This appendix describes how to compute the homographies between the different views using the given point correspondences.

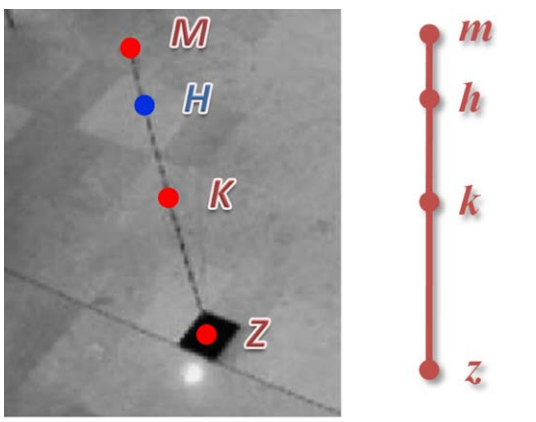


Fig. 11 Detected points along a pole

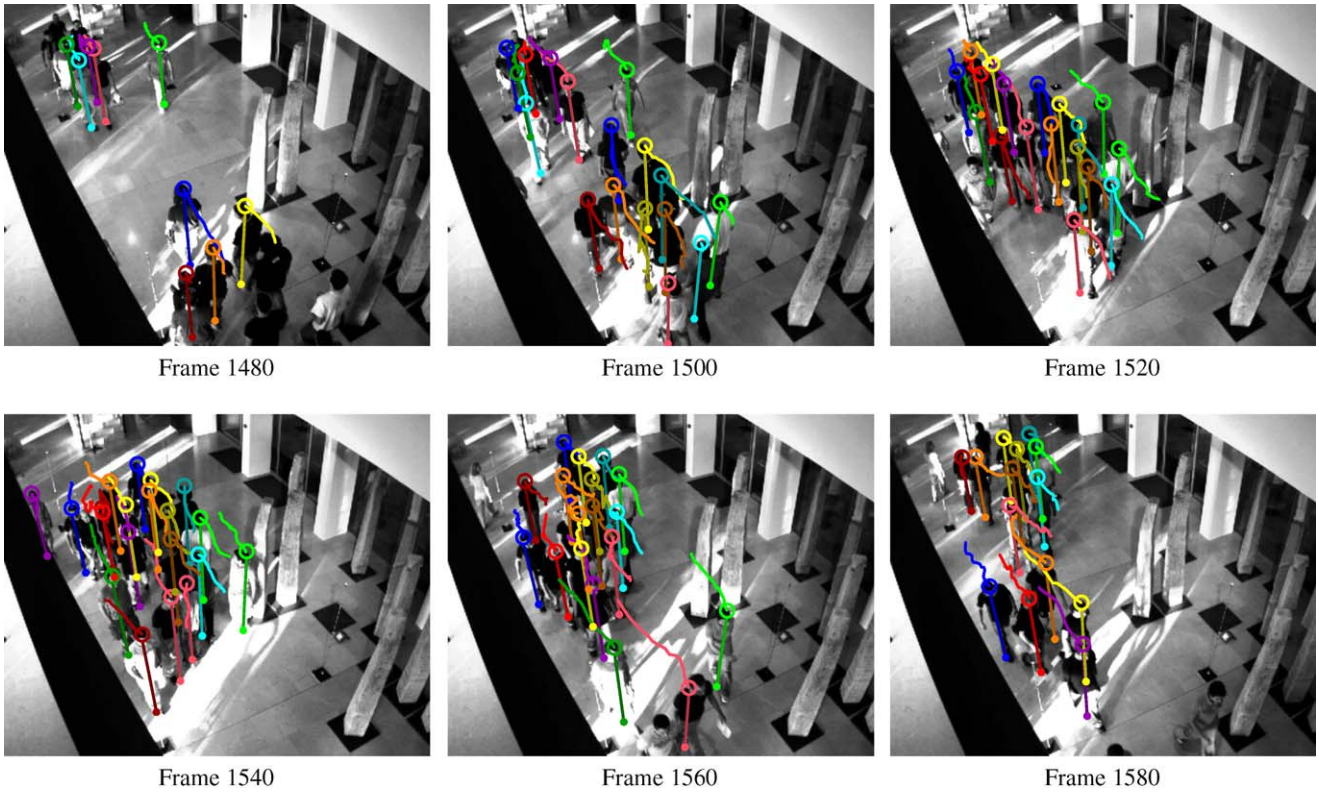


Fig. 12 Selected frames from sequence S3c

In order to compute a homography between two views, four corresponding points on a plane are required. For our algorithm, only planes parallel to the ground are used. Our setup consists of four vertical poles placed in the scene, with three points at known heights on each of them. We next show how from the projection of these points to an image, the projection of new points along the poles at any given height can be found. Using these new points, the required homography can be computed.

Let z, k, and m be the heights of the three points along a pole (on the bottom, middle and top of the pole, respectively), and let their projections to the image plane be Z, K and M (see Fig. 11). The projection, H, of a new point at height h along the pole can be computed using the observation that the cross-ratio of the four scene points is equal to the cross-ratio of their projections.

In world coordinates, all four points are known (located at known heights along the pole), and therefore their cross-ratio can be computed (all values are scalar):

$$r = \frac{(h-z)(m-k)}{(h-k)(m-z)} \tag{A.1}$$

Since cross-ratio is preserved by perspective projection, we can write the same equation for the distances between the image points:

$$r = \frac{HZ \cdot MK}{MZ \cdot HK} \tag{A.2}$$

where HZ denotes the distance between points H and Z on the image plane.

In the above equation, r is known, but HZ and HK are not. Since $HK = HZ - KZ$, we can replace HK, and remain with a single unknown parameter, HZ:

$$r = \frac{HZ \cdot MK}{MZ \cdot (HZ - KZ)} \tag{A.3}$$

From this, HZ can be extracted:

$$HZ = \frac{r \cdot MZ \cdot KZ}{r \cdot MZ - MK} \tag{A.4}$$

Repeating this process for each of the four poles, four points on the plane parallel to the ground at height h can be obtained. From these points, the required homography can be computed (using the Direct Linear Transformation algorithm, as described in Hartley and Zisserman (2000)).

References

Ali, S., & Shah, M. (2008). Floor fields for tracking in high density crowd scenes. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 1–14).

Arsic, D., Hristov, E., Lehment, N., Hornler, B., Schuller, B., & Rigoll, G. (2008). Applying multi layer homography for multi camera person tracking. In *International conference on distributed smart cameras*.

Brostow, G. J., & Cipolla, R. (2006). Unsupervised Bayesian detection of independent motion in crowds. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 594–601).

Cai, Q., & Aggarwal, J. K. (1999). Tracking human motion in structured environments using a distributed-camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(11), 1241–1247.

Cheung, G. K. M., Kanade, T., Bouguet, J. Y., & Holler, M. (2000). A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 714–720).

Du, W., & Piater, J. H. (2007). Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In *Proceedings of the Asian conference on computer vision (ACCV)* (pp. 365–374).

Eshel, R., & Moses, Y. (2008). Homography based multiple camera detection and tracking of people in a dense crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Faugeras, O. D. (1993). *Three-dimensional computer vision*. Boston: MIT Press.

Felzenszwalb, P. F. (2001). Learning models for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 56–62).

Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2007). Multi-camera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*.

Franco, J. S., & Boyer, E. (2005). Fusion of multi-view silhouette cues using a space occupancy grid. *Proceedings of the International Conference on Computer Vision*, *2*, 1747–1753.

Garibotto, G., & Cibei, C. (2005). 3D scene analysis by real-time stereovision. In *Proceedings of the international conference on image processing (ICIP)* (pp. 105–108).

Gavrila, D. M., & Philomin, V. (1999). Real-time object detection for smart vehicles. In *Proceedings of the international conference on computer vision (ICCV)* (pp. 87–93).

Goldschmidt, R., & Moses, Y. (2008). Practical calibration and synchronization in a wide baseline multi-camera setup using blinking LEDs. Technical report. Interdisciplinary Center Herzliya ftp://ftp.idc.ac.il/Pub/Users/cs/yael/TR-2008/IDC-CS-TR-200801.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Isard, M., & MacCormick, J. (2001). BraMBLe: a Bayesian multiple-blob tracker. In *Proceedings of the international conference on computer vision (ICCV)* (pp. 34–41).

Kettnaker, V., & Zabih, R. (1999). Bayesian multi-camera surveillance. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 253–259).

Khan, S. M., & Shah, M. (2006). A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of the European conference on computer vision (ECCV)* (pp. IV: 133–146).

Khan, S. M., Yan, P., & Shah, M. (2007). A homographic framework for the fusion of multi-view silhouettes. In *Proceedings of the international conference on computer vision (ICCV)*.

Kim, K., & Davis, L. S. (2006). Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 98–109).

Kobayashi, Y., Sugimura, D., Hirasawa, K., Suzuki, N., Kage, H., Sato, Y., & Sugimoto, A. (2006). 3D head tracking using the particle filter with cascaded classifiers. In *Proceedings of the British machine vision conference (BMVC)* (pp. I:37).

Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., & Shafer, S. (2000). Multi-camera multi-person tracking for easyliving. In *International workshop on visual surveillance*.

Laurentini, A. (1994). The visual hull concept for Silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(2), 150–162.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 878–885). Washington: IEEE Computer Society.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–136.

Mittal, A., & Davis, L. (2001). Unified multi-camera detection and tracking using region matching. In *Proceedings of the IEEE workshop on multi-object tracking*.

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

Nummiaro, K., Koller Meier, E., Svoboda, T., Roth, D., & Van Gool, L. J. (2003). Color-based object tracking in multi-camera environments. In *German pattern recognition symposium* (pp. 591–599).

Orwell, J., Remagnino, P., & Jones, G. A. (1999). Multi-camera color tracking. In *Proceedings of the IEEE workshop on visual surveillance* (p. 14).

Papageorgiou, C., & Poggio, T. (1998). Trainable pedestrian detection. In *IEEE conference on intelligent vehicles* (pp. 35–39).

Polana, R., & Nelson, R. (1994). Low level recognition of human motion (or how to get your man without finding his body parts). In *Proceedings of the IEEE workshop on motion of non-rigid and articulated objects* (pp. 77–82).

Quaritsch, M., Kreuzthaler, M., Rinner, B., Bischof, H., & Strobl, B. (2007). Autonomous multicamera tracking on embedded smart cameras. *Journal on Embedded Systems*, *2007*, 10.

Rodriguez, M. D., & Shah, M. (2007). Detecting and segmenting humans in crowded scenes. In *Proceedings of the international conference on multimedia* (pp. 353–356).

Shashua, A., Gdalyahu, Y., & Hayun, G. (2004). Pedestrian detection for driving assistance systems: single-frame classification. In *IEEE conference on intelligent vehicles* (pp. 1–6).

Smith, K., Gatica-Perez, D., & Odobez, J. (2005). Using particles to track varying numbers of interacting people. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 962–969).

Viola, P. A., Jones, M. J., & Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, *63*(2), 153–161.

Wu, B., & Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, *75*(2), 247–266.

Yilmaz, A., Javed, O., & Shah, S. (2006). Object tracking: a survey. *ACM Journal of Computing Surveys*, *38*(4), 13.

Yu, Q., Medioni, G., & Cohen, I. (2007). Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Zhao, T., & Nevatia, R. (2004). Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1208–1221.