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

M.Sc. dissertation for research project

Submitted by Ran Eshel
Under the supervision of Dr. Yael Moses

September 2008
Acknowledgments

I would like to express my appreciation to my supervisor, Dr. Yael Moses, for her support and encouragement during the two years I worked on this thesis. Her broad knowledge and analytical way of thinking have been of great value to me, while her focused and practical approach helped to keep me on the right track. I could not have completed this work without her endless patience and dedication.

I would like to thank the founding dean of the Efi Arazi School of Computer Science, Professor Shimon Schocken, for having faith in me when I did not, and the current dean, Professor Gadi Taubenfeld, for his continued support.

I would also like to thank the wonderful teachers I had here at IDC, who reminded me why I had chosen this vocation, and the other faculty members with whom I had the pleasure of working: Professor Yakov Hel-Or, Dr. Ariel Shamir, Professor Noam Nissan, Dr. Tami Tamir and Dr. Anat Bremler-Barr.

Finally, I would like to thank my fellow students: Ran Goldschmidt – for participating in our experiments and assisting in data capture, and for developing the algorithm and writing the code used for calibration and synchronization of the test sequences; and Nir Chen – for asking interesting questions, for sharing thoughts and ideas, and for the many hours we spent together in the lab.

This research was supported by the Israel Science Foundation (grant no. 1339/05).
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 thesis 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 m²), in spite of severe and persistent occlusions.
# Table of Contents

Acknowledgments ................................................................. i
Abstract .................................................................................. ii
Table of Contents ................................................................. iii
List of Figures ......................................................................... v

1 Introduction ............................................................................. 1

2 Related Work ......................................................................... 5
   2.1 Single Camera Approaches ................................................... 5
   2.2 Multiple Camera Approaches ............................................... 7
   2.3 Homography-Based Multiple Camera Approaches ..................... 9

3 The Method ............................................................................. 10
   3.1 Head Top Detection ............................................................. 10
      3.1.1 2D Patch Detection ....................................................... 10
      3.1.2 Finding the Highest 2D Patch ........................................ 13
      3.1.3 Expected Problems ...................................................... 16
   3.2 Tracking ............................................................................. 16

4 Experimental Results ............................................................. 20
   4.1 Implementation and System Details ....................................... 20
   4.2 Sequences and Results ....................................................... 21
   4.3 Varying the Number of Cameras .......................................... 28
# TABLE OF CONTENTS

5 Conclusion ................................................. 29
  5.1 Summary ............................................. 29
  5.2 Future Work .......................................... 30

Appendix A: Transforming Between Views .................. 33

Appendix B: Projecting to the Ground Plane .............. 35

Bibliography ................................................. 38
List of Figures

1.1 Six views of a single scene taken at the same time ......................... 2
1.2 Tracking results using combined data from all views .................... 3

3.1 2D patch detection demonstrated on a single person .................... 11
3.2 Using hyper-pixel intensity variance to prevent false positive detections ................................. 12
3.3 Intermediate results of head top detection .............................. 14
3.4 Match likelihood score parameters ................................. 18

4.1 Examples of tracked trajectories from four sequences ................................. 22
4.2 Selected frames from sequence S3c .................................. 25
4.3 Selected frames from sequence S4 .................................. 26
4.4 Selected frames from outdoor sequence S1 ............................... 27
4.5 System performance as a function of the number of cameras ............... 28
Chapter 1

Introduction

People tracking has emerged as an important research topic in the computer vision community, mainly, but not exclusively, for surveillance applications. As a result of the growing availability of low cost image sensors, the increase in network bandwidth, and the advances in storage technology, systems comprising tens or hundreds of video cameras have become ubiquitous. Since this process could not be accompanied by a similar increase in manpower, a single human operator is now commonly required to monitor an impossibly large number of cameras at once. Consequently, much of the available video data is ignored in real-time, and merely recorded for future reference.

The only way to utilize these large networks of cameras to their full potential is by using an automated visual surveillance system, that can handle large amounts of video data in real-time, and robustly track all of the people in the area of interest. Although many such systems have been suggested by researchers, and some even offered as commercial products, none has been able to fully address realistic surveillance scenarios, such as operating in a crowded airport, train station, or shopping mall. These methods perform well in simpler domains, for example estimating the number of people in a queue, or sounding an alarm when an object is detected in a restricted area, but do not provide a comprehensive solution to the general tracking problem.

The main challenge encountered by tracking methods is the severe and persistent occlusion prevalent in images of a dense crowd (as shown in Figure 1.1). 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 [33] or Zhao and Nevatia [40] 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 [26] or Krumm et al. [22]).

In this thesis 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 almost 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 Figure 3.3). 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 Figure 1.1).

We rely on the fact 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 \[8\] 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 Figure 3.1 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 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 Figure 1.2.

Our 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 $m^2$).

The rest of the thesis is organized as follows: in the next chapter we present a review of previous work. Chapter 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 Chapter 4. Finally, in Chapter 5, we discuss these results, and suggest future research directions.
Chapter 2

Related Work

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. Papageorgiou and Poggio [29] use SVM detectors based on Haar wavelets to detect pedestrians. Gavrila and Philomin [13] construct a template hierarchy from example shapes, with oriented edges as features. Images are then matched against templates by traversing the template hierarchy in a coarse-to-fine manner. Conversely, Felzenszwalb [9] uses learning to construct a single human shape model which captures the information from all examples. People are detected by applying edge detection to the image, and then minimizing the Hausdorff distance between candidate shapes and the learned model. All of these methods are trained on full human figures, and will not perform well if subjects are even partially occluded. More recent methods achieve improved performance by concentrating on local features, or by combining them with global ones.

Shashua, Gdalyahu and Hayun [34] try to reduce the class variability by breaking down regions of interest into sub-regions that correspond to local features, and by dividing the training set into clusters which represent specific poses or illumination conditions. Leibe, Seemann and Schiele [24] use a full-body representation, but increase its flexibility by allowing inter-
polation between local parts seen on different training objects. To enforce global consistency, the hypotheses are then refined with global shape cues, retrieved by matching trained shape templates to image content. Wu and Nevatia [37] detect body parts by boosting a number of weak classifiers based on edgelet features, which are then combined to form a joint likelihood model. The model includes an analysis of possible occlusions. Tracking of partially occluded humans is achieved using data association and mean shift. Dalal and Triggs [6] compared several types of features sets for human detection, and concluded that a very high detection rate can be achieved using Histograms of Oriented Gradient. However, while these and other 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. Polana and Nelson [30] search for repetitive motion, and match against a spatio-temporal template of motion features. If a match is found, it can be used to segment, track and recognize a person without the need to identify specific body parts. Bobick and Davis [2] expand this concept by using motion history images, which more precisely capture the nature of the motion. On the other hand, this makes their method more susceptible to occlusion, and it is therefore not used for tracking. Brostow and Cipolla. [3] track simple image features and probabilistically group them into clusters representing independently moving entities. Space-time proximity and trajectory coherence are used as the only criteria for clustering, and therefore no subject-specific model is required. 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, Jones and Snow [36] improve on previous results by integrating motion information with appearance information. AdaBoost is used to construct a classifier from a heterogeneous set of features, which includes appearance filters and various types of motion filters. 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 methods employ a Bayesian framework, using Kalman filters or particle filters for tracking: Isard and MacCormick [17] handle occlusions using a 3D object model that provides depth ordering, and apply the CONDENSATION algorithm [16] for tracking multiple people, whose number may vary during tracking; Zhao and Nevatia [40] use a coarse 3D human shape model to separate between different people that belong to a single foreground blob; Smith et al. [35] and Yu et al. [39] use sophisticated background subtraction methods for detection, and an MCMC approach to sample the solution space efficiently.

These and other single camera methods are inadequate for handling highly dense crowds such as those considered in this thesis, due to severe occlusion which results in large foreground blobs 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, Zhao and Nevatia could not be performed, since their method was reported to be inapplicable under these challenging density and illumination conditions. ¹

Rabaud and Belongie [32] suggest a method that operates well in highly dense crowds, by detecting a rich set of features, without performing background subtraction. These features are then clustered based on local rigidity constraints. However, their method only provides a rough count of people, and does not perform full tracking.

### 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 try 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 transferred from one camera to another [4, 18, 31]. 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 over-

¹ Personal communication.
lapping fields of view. Kobayashi et al. [21] and Nummiaro et al. [27] use multiple cameras to robustly track a single target, but most multiple camera methods attempt to negotiate more challenging scenarios.

Krumm et al. [22] use pairs of cameras to resolve ambiguity using 3D stereo information. Their method is based on background subtraction, and is hence limited when a dense crowd is considered. Orwell, Remagnino and Jones [28] 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 [26] 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.

Du and Piater [7] 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. Fleuret et al. [10] 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 Gariboto and Cibei [12]. 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 [19]. They use a homography transformation to align the foreground of the ground plane from images taken from a set of cameras with overlapping fields of view, and achieve good results in moderately crowded scenes. 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. However, their method seems inadequate for handling highly crowded scenes. On one hand, tracking people’s feet rather than their heads precludes the use of intensity value correlation, since the 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 Section 3.1.3, and demonstrated in Figure 3.3b).

Recently, Khan, Yan and Shah [20] suggested applying the same concept to planes at multiple heights 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 [23]), or on constructing a space occupancy grid (Cheung et al. [5], Franco and Boyer [11]). 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, Javed and Shah [38].
Chapter 3

The Method

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.

We assume a set of synchronized and partially calibrated cameras overlooking a single scene, where head tops are visible. The partial calibration consists of the correspondence of 12 points between all views. From this, homographies of a plane parallel to the floor at any height and between any two views can be generated.

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 3.3 shows intermediate results of the method described below.

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 Figure 3.1 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 \( \pi^h \) be a plane in the 3D scene parallel to the floor at height \( h \). A \( \pi \)-mapping between an image and a reference image is defined as the homography that aligns the projection of points on the plane \( \pi \) in the two images. For a plane \( \pi^h \) and sequences \( S_i \) and \( S_1 \), it is given by the \( 3 \times 3 \) homography matrix \( A_{i,1}^h \). Using the correspondences given by the partial calibration, the homography matrices \( A_{i,1}^h \) can be computed for any height \( h \) (see Appendix A).
Figure 3.2: 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.

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 $A_{i,1}^h$. 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 $\pi^h$-mapped to $q \in S_1(t)$. The $\pi^h$-mapping of the point $p \in S_i(t)$ to a point $q$ in frame $S_1(t)$ is given by $q = A_{i,1}^h p_i$. The inverse transformation, $p_i = A_{1,i}^h q$, allows us to compute $\bar{q}^h$:

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

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 $\pi^h$. 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 \( \pi^h \). 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 \( \pi^h \). 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 [25]. 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 \( H = \{h_1, ..., h_n\} \) of expected people heights. The set is taken at a resolution of 5 centimeters, within the range 150-190 centimeters. 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 \( \pi^h \) in the reference image with the location of its projection to the plane \( \pi^0 \) in the same image. For each height \( h_i \in H \), the homography transformation that maps the projection of the plane \( \pi^{h_i} \) to the floor of sequence \( S_1 \) is given by the \( 3 \times 3 \) homography matrix \( B^{h_i} \). These matrices
Figure 3.3: 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.

can be computed based on the partial calibration assumption of our system (see Appendix B). 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 \( B^{hi} q \). 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.
Algorithm 3.1 2D patch detection at time $t$

```plaintext
foreach image $S_i(t)$ do
    Detect foreground pixels using background subtraction
end for

for $h = h_{top}$ to $h_{bottom}$ 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 $\bar{q}^h$
            $\bar{q}^h(i) \leftarrow I_i(A_{1,i}^h,q)$
        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
    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 list of head centers for height $h$
    end for
end for

for $h = h_{top}$ to $h_{bottom}$ do
    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 $c_i \in C(t)^{h'}$ such that $h' < h$ and $\|c_i - c_j\| \leq \text{thresh}$
    end for
end for
return $L(t)$
```
3.1.3 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 tops. Figure 3.2b 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 [19], such accidental detections will occur even if people are wearing different colors (as in Figure 3.2c). 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 (see Section 4.3). We remove phantoms 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 thesis.

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 expected region, the nearest neighbor is chosen. If no feature is found within the 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 numbers 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:

\[
\text{overlap}(tr_i, tr_j) = f(tr_j) - \ell(tr_i)
\]

Two tracks, \( tr_i \) and \( tr_j \), are considered for merging if:

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

A \textit{match likelihood score} is computed for each pair of tracks that satisfies this condition. The score is a function of the following measures:

- \( m_1 \) – the number of overlapping frames 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 \( tr_j \)
- \( m_5 \) – the minimal distance between corresponding points along the overlapping segments (or along the expected paths of the trajectories, in case of a negative overlap)
- \( m_6 \) – the average distance between corresponding points along the overlapping segments
Figure 3.4: 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 average distance between corresponding points along the overlapping segments; $m_6$ – the minimal distance between corresponding points (in this example, the points designated 3 in each track).

The match likelihood score is defined by:

$$score(tr_i, tr_j) = \frac{1}{6} \sum m_i/\hat{m_i}$$

where $\hat{m_i}$ 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 weighted components which include the track length, and the average change in speed, direction and height between any two consecutive time steps. 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.
Chapter 4

Experimental Results

To demonstrate the effectiveness of our method, we performed experiments on real video sequences under changing conditions. In Section 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 Section 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 meters, viewing the scene at a relatively sharp angle (45° or more below the horizon). Detection and tracking were performed on an area of $3 \times 6$ meters. All test sequences were taken at a rate of 15 frames per second, with an image size of $640 \times 512$.

The cameras were calibrated using a novel method described in Goldschmidt and Moses [14]. 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 A). The same data is also used to synchronize the sequences, and to compute the ground plane projection homography matrices, $B^h$ (see Appendix B).

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.

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 (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 4.1 summarizes the tracking results. Examples can be seen in Figure 1.2 and in Figure 4.1, where each detected person is marked by his head center. The tails mark the
Figure 4.1: 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.

detected trajectories up to the displayed frame.

We next describe each sequence in detail:\(^1\)

**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

\(^1\)Tracking results can be seen in: ftp://ftp.idc.ac.il/Pub/Users/CS/Yael/CVPR-2008/CVPR-2008-results.zip
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 4.1a presents the tracking results on this sequence.

**S2:** A 1100 frame long 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 (200, 250 and 300 frames long) from an indoor sequence 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** 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. Despite similar density, tracking results are significantly better than in sequence **S3**, 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). Figure 4.1b presents the tracking results on sequence **S3b**. Figure 1.2 and Figure 4.2 present the tracking results on sequence **S3c**.

**S4:** A high crowd density sequence (200 frames), taken using 6 cameras placed around the
Chapter 4 / Experimental Results

### Table 4.1:

<table>
<thead>
<tr>
<th>Sequence</th>
<th># Cams</th>
<th>GT</th>
<th>TP</th>
<th>PTP</th>
<th>IDC</th>
<th>DR %</th>
<th>FN</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6</td>
<td>27</td>
<td>26</td>
<td>23</td>
<td>3</td>
<td>98.7</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>S2</td>
<td>9</td>
<td>42</td>
<td>41</td>
<td>39</td>
<td>0</td>
<td>97.9</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>S3a</td>
<td>9</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S3b</td>
<td>9</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>S3c</td>
<td>9</td>
<td>21</td>
<td>21</td>
<td>20</td>
<td>1</td>
<td>99.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>6</td>
<td>23</td>
<td>23</td>
<td>22</td>
<td>0</td>
<td>99.1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S5</td>
<td>6</td>
<td>24</td>
<td>23</td>
<td>14</td>
<td>12</td>
<td>94.4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>174</strong></td>
<td><strong>171</strong></td>
<td><strong>155</strong></td>
<td><strong>16</strong></td>
<td><strong>98.4</strong></td>
<td><strong>3</strong></td>
<td><strong>14</strong></td>
<td></td>
</tr>
</tbody>
</table>

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 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 4.1c and Figure 4.3 present the tracking results on this sequence.

**S5:** A very 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 $8\text{m}^2$ area, a density of over 2.5 people per $\text{m}^2$. 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 4.1d presents the tracking results on this sequence. The tails demonstrate the complex motion of the people.
Figure 4.2: Selected frames from sequence S3c.
Chapter 4 / Experimental Results

Figure 4.3: Selected frames from sequence S4.
Figure 4.4: Selected frames from outdoor sequence S1.
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 Figure 4.5. 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 Section 3.1 and demonstrated in Figure 3.2b, decreasing the number of cameras may increase the number of accidental matchings, causing phantoms to appear. The effect of this phenomenon is apparent in Figure 4.5b. 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.
Chapter 5

Conclusion

5.1 Summary

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 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.
5.2 Future Work

Since this work focuses on a relatively new approach towards human detection, it can be extended in many ways:

Detection:
The main fault of the detector is false positives, which we refer to as 'phantoms' (see Section 3.1.3). Most of these are random and inconsistent, and can thus be easily removed by the tracker. Others, however, are more persistent, and cannot be identified based on motion characteristics alone. To remove these phantoms, we suggest augmenting our detector with a human shape detector, which will verify that each detected object is indeed a person.

Tracking:
Currently, tracking is based solely on the positions of the different objects as found by the tracker. This is adequate when the tracking quality is high, and the objects are far apart, but may result in tracking errors, such as ID changes or lost tracks, in more challenging scenarios. Object recognition may assist in preventing these failures - using characteristics such as color signature, specific objects can be identified and reliably tracked even when surrounded by other objects.

It may also be interesting to compare the heuristics-based tracker used in this work with a more conventional tracker based on particle filters. While the former has proved itself to be quite robust in the highly dense crowds considered here, the latter is becoming the de facto standard, and may produce better results.

Detector / Tracker Interface:
The process of passing information from the detector to the tracker involves significant data loss. The variance information for each pixel is thresholded, and only values above the threshold are passed to the tracker as candidate objects for tracking. The effect of this problem is somewhat diminished by the use of two different thresholds (see Section 3.2), but this approach should be expanded - instead of passing lists of possible locations to the tracker, Bayesian reasoning can be used to generate a probability density function for the whole image. This
will provide the tracker with a continuous rather than a discrete assessment of whether or not an object is present in each specific pixel.

Another possible improvement could be to use the detection results from previous frames as a prior to the detector in the current frame. Alternately, data can be taken from the results of the tracking phase up until the current frame, thus reducing errors that might occur due to the noisy output of the detection phase.

**Prior Knowledge:**
The algorithm described here does not use any knowledge of the scene structure for track initialization. Therefore, tracks may appear or disappear at any point in the image. Many false positive tracks may be removed simply by constraining the area in which new objects may enter the scene based on real-world data. Similarly, the number of objects lost by the tracker can be reduced if tracks are only allowed to disappear within a given area.

**Parameters:**
When using the described algorithm, several parameters must be set based on the operating environment. These include the expected head size, the maximum expected speed, the expected range of human heights, and several thresholds. To improve the usability and the robustness of the algorithm, most of these parameters should be removed: for some, it is possible to find a single universal value that need not be changed; most of the others can be set automatically, based on an analysis of the given scene.

**Performance:**
The current implementation of the algorithm performs object detection and tracking with very high accuracy, but its performance is far from real-time. This can be improved by optimizing performance bottlenecks, such as the homography calculation process, or by translating the whole implementation from Matlab to a more efficient language, such as C/C++.

A more radical approach towards improving the algorithm’s performance involves distributed computing – instead of cameras that merely perform data collection, smart cameras may be used, utilizing the highly parallelizable nature of the algorithm. Most of the algorithm’s
operations are done in each view separately, therefore if each camera has its own computing power, these operations can be performed in parallel, and only the results sent to the central computer.

Testing:
The results described above (see Table 4.1) provide an assessment of the algorithm’s performance on our own test data, but give no direct comparison to other multiple-camera algorithms. This is because we could not find any standard set of movies that fits our scenario (cameras with overlapping fields of view), and on which other algorithms were tested. Indeed, from examining the existing literature, it appears that most multiple-camera algorithms were only tested on the author’s own data. It seems essential that such a set be created and placed in public domain, so that the growing number of multiple-camera algorithms can be directly and accurately compared with each other.
Appendix A

Transforming Between Views

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

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 computed.

Let \( z \), \( k \), and \( m \) be the heights of the three points along a pole, and let their projections to the image plane be \( Z \), \( K \) and \( M \) (on the bottom, middle and top of the pole, respectively). 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):

\[
r = \frac{(h - z)(m - k)}{(h - k)(m - z)}
\]  

(A.1)

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

\[ r = \frac{HZ \cdot MK}{MZ \cdot HK} \]  \hspace{1cm} (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, \( HZ \):

\[ r = \frac{HZ \cdot MK}{MZ \cdot (HZ - KZ)} \]  \hspace{1cm} (A.3)

From this, \( HZ \) can be extracted:

\[ HZ = \frac{r \cdot MZ \cdot KZ}{r \cdot MZ - MK} \]  \hspace{1cm} (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 Hartely and Zisserman [15]).
Appendix B

Projecting to the Ground Plane

Combining results from different heights requires projecting 2D patches to the ground plane. In this appendix we prove that similarly to projecting between views, projecting a point vertically to a plane at a different height in the same view can also be done using homography. Instead of matching a point as seen from two different views, we match points at different heights as seen from a single view. From these matches, a homography from the plane at height $h$ to the ground plane can be computed, again, using Hartely and Zisserman’s DLT algorithm [15].

We next show that such a homography exists:

**Claim.** Any two planes parallel to the ground at heights $K$ and $L$ are related by homography transformation.

**Proof.** Given a point $(X, Y, Z, 1)^T$ on the plane at height $K$, its projection to the image plane in homogeneous coordinates is given by:

$$
\begin{pmatrix}
X \\
K \\
Z \\
1
\end{pmatrix}
= 
\begin{pmatrix}
x \\
y \\
w
\end{pmatrix}
$$

(B.1)
Appendix B: Projecting to the Ground Plane

The coordinates $x$, $y$, $w$ can be computed as:

$$
\begin{align*}
  x &= M_{11}X + M_{12}K + M_{13}Z + M_{14} \\
  y &= M_{21}X + M_{22}K + M_{23}Z + M_{24} \\
  w &= M_{31}X + M_{32}K + M_{33}Z + M_{34}
\end{align*}
$$

(B.2)

This transformation can be represented as a $3 \times 3$ matrix:

$$
P_K = \begin{pmatrix}
  M_{11} & M_{13} & M_{12}K + M_{14} \\
  M_{11} & M_{13} & M_{22}K + M_{24} \\
  M_{11} & M_{13} & M_{32}K + M_{34}
\end{pmatrix}
$$

(B.3)

Thus, for any point on the plane parallel to the ground plane at height $K$, its projection to the image plane is given by:

$$
P_K \begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ w \end{pmatrix}
$$

(B.4)

Similarly, a point with the same $X$, $Z$ coordinates, but at height $L$, will be projected to a different point $x'$, $y'$, $w'$ on the image plane:

$$
P_L \begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix}
$$

(B.5)

Since the projection matrix $P_K$ is non-singular, it can inverted and thus (B.4) can be modified to retrieve the point $(X, Z, 1)^T$ on the plane at height $K$ from its projection to the image plane:

$$
\begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} = P_K^{-1} \begin{pmatrix} x \\ y \\ w \end{pmatrix}
$$

(B.6)
Combining equations (B.5) and (B.6), we get the homography $H_{K,L}$ that projects a point from height $K$ to height $L$:

\[
\begin{pmatrix}
x' \\
y' \\
w'
\end{pmatrix} = P_L \cdot P_K^{-1} \begin{pmatrix} x \\ y \\ w \end{pmatrix} = H_{K,L} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \quad (B.7)
\]

Setting $K$ to the head height $h$, and $L = 0$, we get a homography $H_{h,0}$ for vertically projecting any point at height $h$ to its ground plane location.
Bibliography


