Multi-target Tracking in Time-lapse Video Forensics

Paul Koppen
Intelligent Systems Lab Amsterdam
University of Amsterdam
Science Park 107
1098XG Amsterdam
The Netherlands
w.p.koppen@uva.nl

Marcel Worring
Intelligent Systems Lab Amsterdam
University of Amsterdam
Science Park 107
1098XG Amsterdam
The Netherlands
m.worring@uva.nl

ABSTRACT

To help an officer to efficiently review many hours of surveillance recordings, we develop a system of automated video analysis. We introduce a multi-target tracking algorithm that operates on recorded video. Apart from being robust to visual challenges (like partial and full occlusion, variation in illumination and camera view), our algorithm is also robust to temporal challenges, *i.e.*, unknown variation in frame rate. The complication with variation in frame rate is that it invalidates motion estimation. As such, tracking algorithms that are based on motion models will show decreased performance. On the other hand, appearance based tracking suffers from a plethora of false detections. Our tracking algorithm, albeit relying on appearance based detection, deals robustly with the caveats of both approaches. The solution rests on the fact that we can make fully informed choices; not only based on preceding, but also based on following frames. It works as follows. We assume an object detection algorithm that is able to detect all target objects that are present in each frame. From this we build a graph structure. The detections form the graph's nodes. The vertices are formed by connecting each detection in one frame to all detections in the following frame. Thus, each path through the graph shows some particular selection of successive object detections. Object tracking is then reformulated as a heuristic search for optimal paths, where optimal means to find all detections belonging to a single object and excluding any other detection. We show that this approach, without an explicit motion model, is robust to both the visual and temporal challenges.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*motion, perceptual reasoning, representations, data structures, and transforms, video analysis*

General Terms

Algorithms, Security

1. INTRODUCTION

The goal of this work is to aid a reviewer in the task of processing large amounts of incriminating video in forensic cases. In such cases, there is generally an overload of video available of which only a fraction is of interest to the reviewer. The computer could help to speed up the tedious process, if it could provide information on all the recorded persons. Information like when they entered the scene and with whom.

Because all of the available video is of potential interest, we cannot set any conditions on the video to be processed. The video may well be of poor quality, is hindered by weather conditions, and the camera may have zoomed and panned by an operator. Surveillance video is also mostly time-lapse, meaning that the frame rate is dynamic (to save bandwidth).

Currently, tracking algorithms ([2, 12, 7, 11] to name a few), each focusing on specific attributes particular to their application, assume that objects show limited movement between two successive frames. Although true in many situations, this assumption is particularly invalid in time-lapse video where the interval between two successive frames can be large.

The assumption of limited object movement is closely related to the fact that the algorithms process video in chronological order — tracking objects from one frame to the next. In the domain of forensics, however, the video has previously been recorded and can thus be processed in any desired order. As long as this is faster than human reviewing, in fact, we can exhaustively scan each frame for all present target objects. Tracking is then inferred from matching any set of detections, not limited to only directly connected frames.

In this paper we propose a tracking algorithm based on graph searching. The graph is constructed from person detections. A path through this graph connects detections from multiple video frames and as such resembles a person track. Two apparent advantages arise from this approach; 1. searching the graph is a matter of interpolation (in contrast with chronological processing) and is therefore by itself more robust, and 2. it provides an elegant unified way to deal with occlusion and poor person detections.

The paper is organized as follows. In section 2 we briefly describe the used method for object detection. Then, in section 3 we introduce the graph structure as a basis for tracking, followed by an explanation of the actual tracking

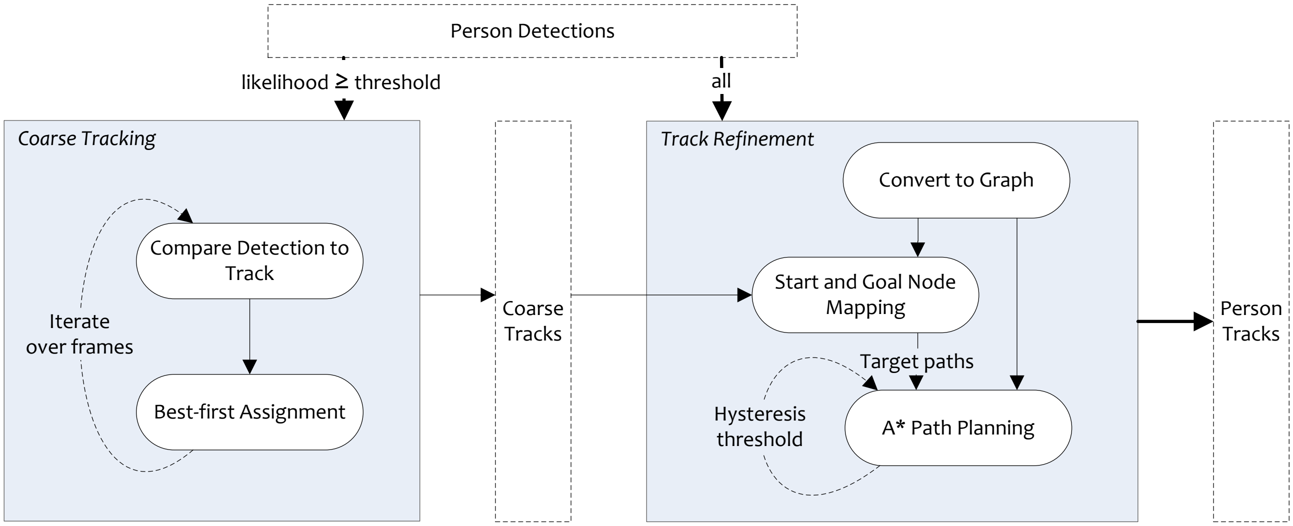

Figure 1: **Tracking framework.** Tracking is performed in two stages. First, coarse tracking runs a simple tracking algorithm on only the best person detections, resulting in accurate but discontinuous tracks. Subsequently, the coarse tracks are refined by searching the graph (which contains all person detections as nodes) for an optimal path.

methods. Section 4 describes the evaluation and the used datasets, with expiremental results presented in section 5. We conclude this paper with a summary in section 6.

2. PERSON DETECTION

We very briefly describe the method used for person detection as it is basically an implementation of the work by [3]. Roughly, the detection is subdivided into three stages; object localization, object description, and person classification.

Object localization is performed by simple frame differencing. Each pixel in the current frame, I_x^t, is compared to its weighed average value over all previous frames B_x^t. That is, $\|I_x^{t+1} - B_x^t\| > \beta$, with β some noise suppressing threshold, marks all 'object pixels'. Bounding boxes are exhaustively fit over the resulting mask and accepted when containing a sufficient fraction of object pixels.

The visual description of a bounding box is calculated using the Histogram of Oriented Gradients [10]. This method models appearance by means of the spatial coherence of gradient information. Patches are sampled and stored in histograms, where the gradient orientation determines the histogram bin and the gradient magnitude determines the quantity that is added. After combining all the histograms following a predefined hierarchical combination, the result is a single vector that describes the object appearance.

Classification of the bounding boxes into human and not human is performed using a SVM, which was trained on the MIT pedestrian database [8] and the INRIA Person dataset [3]. The classification assigns a likelihood value to each bounding box.

Following [9] we set $\beta = 0.05$ and accept person detections if their classification likelihood is above -2.

3. GRAPH-BASED TRACKING

In this section we describe a tracking algorithm that is robust against the challenges of time-lapse surveillance video. To this end we organize the person detections in a graph structure (explained in section 3.1). Step by step the algorithm explores paths in this graph, constituting person tracks. One advantage of this approach is that it is not prone to drift (in the xy plane) because the set of detections is fixed. Because the graph structure explicitly accomodates for skipping frames, our method is particularly robust to occlusion. This is explained in more detail in section 3.2.

Besides occlusion, another reason that a frame is skipped may be simply because detections did not match sufficiently well, or their likelihood value is simply too low. A refinement procedure, with information of both preciding and following frames, will therefore try to fill such gaps. As such, tracking is a matter of *interpolation*, in contrast to any other current tracking method. Track refinement is described in section 3.3.

3.1 From video to graph

The mapping from video to a graph is as follows. The vertices represent person detections, edges define a directed connection between them. Particularly, since each video frame consists of a separate set of detections, the edges connect exactly each detection from one frame to all detections in the single next frame. Using this representation, a person track, $k = \{b_m, \ldots, b_n\}$, is simply a *specific path* in the graph.

So, let $G = (V, E)$ be the complete graph with vertices $V = \{b_1, \ldots\}$ and edges $E = \{(b_i^t, b_j^{t+1}), \ldots\}$, then a person track, $p = (k \cap V, E)$, is an induced subgraph from G.

In case of coarse tracks we must be cautious because E contains no edges to skip frames. We therefore extend V with an *empty vertex* for each frame, ϵ^t. Selecting this node in a path means skipping the particular frame (see figure 2). Please notice that the term *coarse* thus means for a path to hold at least one ϵ^t.

3.2 Coarse tracking

The main target for coarse tracking is to set start and end points for graph searching in track refinement. Rather than finding all detections within each track, the predominant

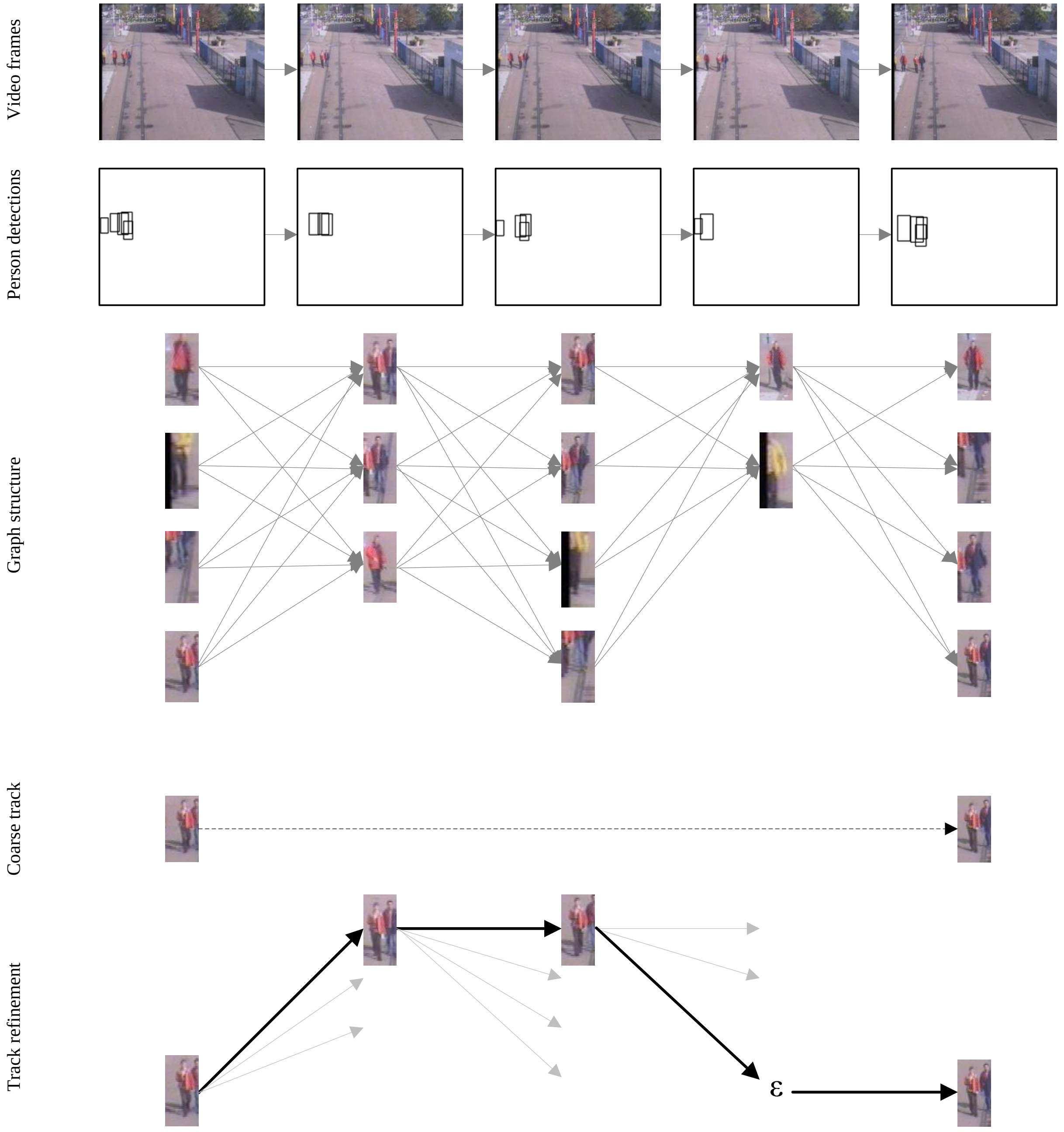

Figure 2: **A graph structure for tracking.** The set of bounding boxes in a frame is represented by a set of vertices, where each vertex is connected to all vertices in the next frame. Task for the tracker is to find the paths that maintain identity —a path where all vertices are of the same person.

criterion for coarse tracking is thus to find at least all person tracks allowing (large) intermediate gaps. Provided that the person detector correctly detects all persons at least a few times, we can put stringent conditions on matching new detections to tracks. At the same time, this also reduces the susceptibility to false detections.

In short, our implementation boils down to a very basic tracking strategy: Starting at frame 1 we iterate over all frames, adding detections to tracks when they match. Only detections with a likelihood score above θ are regarded. Assignment is performed in a best-first fashion and bounded by an upper bound λ, controlling the tightness of matching. Unassigned detections are used to start new tracks.

To refrain from ensuing any bias from person detection, we employ a different visual representation for matching bounding boxes, *i.e.*, not HOG. In this case we describe the appearance using normalized color histograms and compare them using the Bhattacharyya difference measure[1], $D_B(h_a, h_b)$. Matching is performed by comparing the histogram of a detection to both the first and the last histogram in a track. Doing so provides a controlled way of adapting to small appearance changes. Control is determined by a weighing factor, $0 \leq w \leq 1$, which determines the relative importance of matching to either the first or the last histogram.

Summarizing, a bounding box b with a likelihood ratio above θ, is said to match a coarse track, $k = [b_1, \ldots, b_m]$, if

$$\begin{aligned} D_{track}(b,k) = \quad & wD_B(H(b), H(b_1)) + \\ & (1-w)D_B(H(b), H(b_m)) \quad \leq \lambda \end{aligned} \tag{1}$$

with $H(b)$ the histogram of b. Setting $w = 0$ disables any adaptation while $w = 1$ allows to completely drift away from the original object appearance.

3.3 Track refinement

Given some coarse track in the graph G, we define track refinement as the process of searching an alternative, better, path through G where ϵ's are substituted for true person detections in their respective frame. Assuming that for those ϵ's no detections have been found due to low likelihood values, and aided by the context of preceding and following bounding boxes, we perform a heuristic graph search to accurately find the optimal path bridging the gap.

More formally, given a path with some gap $\epsilon^{i+1}, \ldots, \epsilon^{j-1}$, spanned by two detections b^i and b^j, we perform a heuristic search starting at b^i and ending at goal node b^j. The heuristic estimates the distance to goal node b^j, based on their location and appearance. Provided that this heuristic is admissible, the graph search algorithm A* is guaranteed to find the optimal path [5].

A heuristic is admissible if the true cost of a path trough the current node will be at least as large as the estimate. In other words, it should never overestimate the true distance-to-goal. In terms of the location, the Euclidean distance does provide the minimal distance between two points. For appearance, our implementation of the Bhattacharyya measure obeys the triangle inequality and hence is admissible as well.

To assemble both distance measures into one heuristic estimate we assume independence and take the vector length. The independence assumption is not entirely correct because appearance (color) may be structurally influenced by position. As this would lead to the actual path-cost being structurally larger than the estimate, it does not invalidate the admissibility though. So the one heuristic that estimates the distance to the goal node becomes:

$$h(b_i, b_j) = \left\| \begin{array}{c} D_{Eu}(b_i, b_j) \\ D_B(b_i, b_j) \end{array} \right\| \tag{2}$$

4. EVALUATION CRITERIA

In this section we define the evaluation criteria applicable in the context of object tracking. Emphasizing the particular nature of our approach, we introduce slight variations on the general conceptions of precision and recall. We also present the data sets used in our tests.

4.1 Bounding boxes

We define the overlap of a bounding box, b, with a ground-truth box, l, as

$$O(b,l) = \frac{A(b \cap l)}{(A(b) + A(l))/2} \tag{3}$$

where A calculates the surface area of a bounding box. The fraction divides the overlapping surface by the average surface area of the two boxes. It can be seen that it evaluates to a value between 0, for no overlap, and 1, for full overlap and both boxes of the same size. We accept a bounding box as a match if it significantly overlaps some ground-truth box: $O(b,l) \geq 0.7$.

4.2 Person tracks

We notice that track detections can match multiple ground-truth tracks and vice versa. This is, however, undesired because ideally each extracted person track should unambiguously follow a single person throughout its presence in the scene. We will therefore evaluate tracks only against that ground-truth track to which it matches the most bounding boxes (at least two).

Using the above definitions we express performance in terms of precision and recall, which we evaluate at two different levels. At the first level, we count *tracks*. The recall over all tracks is: the fraction of labeled tracks that is at least positively matched once. At the second level we look inside tracks and count *bounding boxes*. The recall for a specific track, T, given the matching ground-truth, L, is: the fraction of bounding boxes in L that are matched by bounding boxes in T. Logically, the precision for a specific track is then the fraction of bounding boxes in T that match a bounding box in L. Notice that we omitted a measure for the overall precision, as this is more a measure of performance on the object detection algorithm.

Although we aim to show results of coarse tracking and the effect of track refinement, it should be noticed that the performance will to a certain extent always depend upon the quality of the person detection. In this case, the total number of bounding boxes generated by person detection is 167,837. 1,965 of them together match 1,936 labels (out of the 42,182). This corresponds to a precision of 0.0117 at maximum recall of 0.0459.

[1] To be precise, although generally referred to as the Bhattacharyya distance, this function is actually the Hellinger distance [1], a variation that, conveniently, obeys the triangular inequality [6].

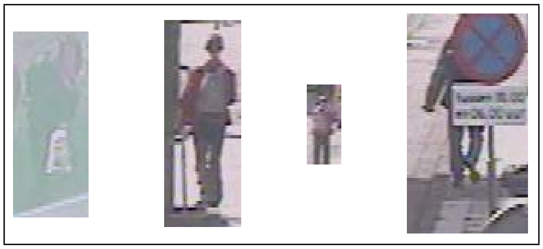

Figure 3: **Examples of ground-truth.** Labeled persons may well be occluded and show large variation is scale. The shown images are cut-outs from the original frame at exactly the bounding box areas.

Video sequence	Frames	Actors	Labels	Tracks
Rotterdam	8,600	4	287	4
BEHAVE	11,200	6	41,895	21
Total	19,800	10	42,182	25

Table 1: **Statistics on the datasets.** A track is bounded by the person being not present for at least 100 frames.

4.3 Test data

There are two data sets used in our tests. The first dataset was acquired in collaboration with the Rotterdam police and comprises video material from actual surveillance cameras in the Rotterdam city center. People walk by naturally, in groups and separately. The images are of reasonable resolution and most of the time quite noisy. The second dataset is the BEHAVE dataset [4]. This dataset is more focused on interaction between people. It mainly shows people flocking together and separating. The quality of the recordings is generally better than the first dataset, although there is less variation in aspects like scene changes, weather, etc. In both datasets the annotations mark persons by means of a bounding box, even if largely occluded. Figure 3 shows some extracted annotations.

5. EXPERIMENTAL RESULTS

In this section we present and discuss the results obtained from the evaluation criteria proposed in this paper. In section 5.1 we evaluate the quality of coarse tracks, particularly in view of providing an optimal basis for track refinement. Section 5.2 then evaluates the effect of refinement on the quality of person tracks.

5.1 Coarse tracking

We assess coarse tracking in its recall and average track precision. The single most important aspect of coarse tracking is its recall, as refinement does not add new tracks. The average track precision provides insight in the priors of refinement. We assess the influence of the upper bound on the bounding-box-to-track distance, λ, and the drift bounding weight, w. Results are shown in figure 4. Notice that, in contrast to common practice, the figures plot both values on the vertical axis. Plotting recall on the horizontal versus precision on the vertical axis would show garbled results for two reasons; the functions are non-monotonic and are not directly related as the precision is calculated over bounding boxes while recall is calculated over tracks.

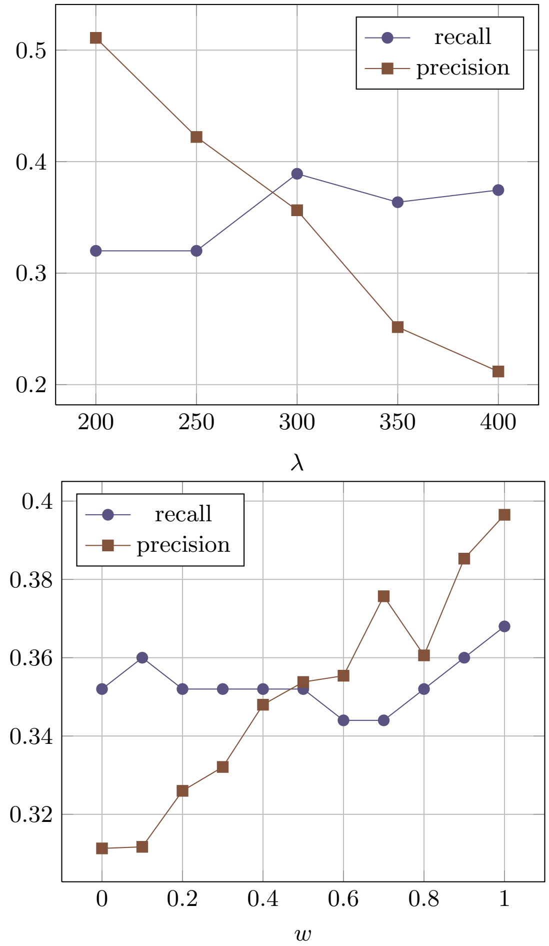

Figure 4: **Coarse tracking results.** Recall of the ground-truth tracks and precision within the detected tracks for varying λ (4a) and w (4b).

The effect of λ is quite intuitive. Relaxing the similarity measure for adding new detections to tracks introduces more false detections, and hence decreases the average track precision. On the other hand, since it adds more bounding boxes in general, it also allows to discover otherwise undetected tracks. In fact, that is happening at $\lambda = 300$ where we notice a sudden increase in recall.

A small surprise lies waiting in parameter w, the ratio of matching new bounding boxes to the first or the last element in a track. Both precision and recall show an optimum at $w = 1$ which means that new bounding boxes should only be matched against the first track element. Any adaptive appearance strategy thus proves to have an adverse effect on performance!

5.2 Track refinement

To control the large amount of noise present in the object detections, track refinement is performed with hysteresis; we repeatedly refine a track, each time decreasing the likelihood threshold, θ, in steps of 0.5. Figure 5 shows the impact of each iteration on the precision and recall. For each plot, the top-left point is the starting point and following resembles a track refinement iteration.

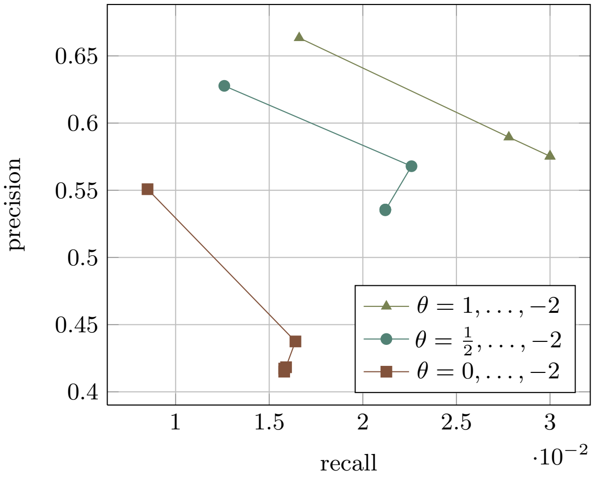

Figure 5: **Track refinement results.** θ decreases in steps of $\frac{1}{2}$. Shown are precision versus recall after each iteration for different starting values of θ (from coarse tracking). The plots show degradation in precision and an increase in recall as an effect of track refinement.

We see that remarkably the second iteration in most cases has also a negative effect on the recall. This can be explained by the fact that tracks that previously did not match any person, after refinement do match a person. Because the recall shows the average recall of each track, and because such added tracks contain at least a few false detections, they are expected to contribute negatively to the recall.

Drawing a line to connect the starting points shows the change in performance when performing pure coarse tracking (without refinement) at different levels of θ. We see here too that the previous notion holds for decreasing recall. More interesting though, is the comparison to the effect of track refinement, which is able to increase the recall while fairly maintaining precision. Coarse tracking at $\theta = 0.5$ plus refining gives a factor 2.7 higher recall *and* increases precision by 3% over simply coarse tracking at $\theta = 0$.

6. SUMMARY AND CONCLUSIONS

In this paper we have proposed a new approach to tracking, which is particularly focused towards video forensics. Differently from the traditional approach, we step away from the real-time and the forward-only constraints. Doing so, our tracking algorithm does not rely on motion estimation and does not need any recalibration. It exploits the fact that the video is off-line to construct a graph to solve the tracking task. We have thoroughly evaluated it on two different datasets, one of which comprises real-life data. Even though the used person detection algorithm did not produce convincing results, we showed that our tracking and refinement algorithm is still able to significantly improve the performance over basic tracking without refinement.

At present we are working on a next massive real-life surveillance dataset and hope to soon be able to evaluate on this dataset too. In the mean time it would be interesting to evaluate the tracking algorithm with a different object detection base.

Acknowledgments

We specially thank Jurrien Bijhold of the Netherlands Forensics Institute for providing the data. Additionally, this work was supported by the EC-FP6 VIDI-Video project and the BSIK MultimediaN project.

7. REFERENCES

[1] G. Chaudhuri. *Bhattacharyya Distance*, chapter 3, pages 121–122. Encyclopaedia of Mathematics. Springer Berlin / Heidelberg, 1997.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. on Comput. Vision and Pattern Recognit.*, volume 2, pages 142–149, Aug. 2000.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Comput. Vision and Pattern Recognit.*, volume 1, pages 886–893, Jun. 2005.

[4] R. Fisher. Behave interactions test case scenarios. Website, Oct. 2007.

[5] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Syst. Sci. and Cybern.*, 4(2):100–107, Jul. 1968.

[6] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. on Commun. Technol.*, 15(1):52–60, Feb. 1967.

[7] H. Nguyen and A. Smeulders. Robust tracking using foreground-background texture discrimination. *Int. J. of Comput. Vision*, 69(3):277–293, Sep. 2006.

[8] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. J. of Comput. Vision*, 38(1):15–33, Jun. 2000.

[9] T. V. Pham, M. Worring, and A. W. M. Smeulders. A multi-camera visual surveillance system for tracking of reoccurrences of people. In *ACM/IEEE Int. Conf. on Distrib. Smart Cameras*, pages 164–169, Sep. 2007.

[10] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Eur. Conf. on Comput. Vision*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002.

[11] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 31(7):1195–1209, July 2009.

[12] W. Zajdel, Z. Zivkovic, and B. Kröse. Keeping track of humans: Have i seen this person before? In *IEEE Int. Conf. on Rob. and Autom.*, pages 2081–2086, April 2005.