Scalable Semi-Automatic Annotation for Multi-Camera Person Tracking

Jorge Niño-Castañeda, Andrés Frías-Velázquez, Nyan Bo Bo, Maarten Slembrouck, Junzhi Guan, Glen Debard, Bart Vanrumste, Tinne Tuytelaars, and Wilfried Philips

***Abstract*— This paper proposes a generic methodology for the semi-automatic generation of reliable position annotations for evaluating multi-camera people-trackers on large video data sets. Most of the annotation data are automatically computed, by estimating a consensus tracking result from multiple existing trackers and people detectors and classifying it as either reliable or not. A small subset of the data, composed of tracks with insufficient reliability, is verified by a human using a simple binary decision task, a process faster than marking the correct person position. The proposed framework is generic and can handle additional trackers. We present results on a data set of ∼6 h captured by 4 cameras, featuring a person in a holiday flat, performing activities such as walking, cooking, eating, cleaning, and watching TV. When aiming for a tracking accuracy of 60 cm, 80% of all video frames are automatically annotated. The annotations for the remaining 20% of the frames were added after human verification of an automatically selected subset of data. This involved ∼2.4 h of manual labor. According to a subsequent comprehensive visual inspection to judge the annotation procedure, we found 99% of the automatically annotated frames to be correct. We provide guidelines on how to apply the proposed methodology to new data sets. We also provide an exploratory study for the multi-target case, applied on the existing and new benchmark video sequences.**

***Index Terms*— Multi-camera tracking, semi-automatic annotation, performance evaluation, people tracking.**

I. Introduction

ONE OF the cornerstone problems in computer vision is that of visual object tracking. Visual tracking has been studied for years, which has resulted in a rich and abundant scientific literature, summarized well in recent surveys [1]–[4].

Manuscript received April 15, 2015; revised September 10, 2015 and February 22, 2016; accepted February 25, 2016. Date of publication March 29, 2016; date of current version April 7, 2016. This work was supported in part by the Flemish Fund for Scientific Research under Grant G0398111N and in part by the Agency for Innovation by Science and Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitrios Tzovaras.

J. Niño-Castañeda, A. Frías-Velázquez, N. Bo Bo, M. Slembrouck, J. Guan, and W. Philips are with the Image Processing and Interpretation Research Group, Ghent University/iMinds, Ghent 9000, Belgium (e-mail: jorge.nino@telin.ugent.be; andres.friasvelazquez@telin.ugent.be; nyan.bobo@telin.ugent.be; mslembro@telin.ugent.be; junzhi.guan@telin.ugent.be; philips@telin.ugent.be).

G. Debard is with MOBILAB, Thomas More Kempen, Geel 2440, Belgium (e-mail: glen.debard@thomasmore.be).

B. Vanrumste is with the Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven/iMinds, Leuven 3001, Belgium (e-mail: bart.vanrumste@kuleuven.be).

T. Tuytelaars is with the Center for Processing Speech and Images, KU Leuven/iMinds, Leuven 3001, Belgium (e-mail: tinne.tuytelaars@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2542021

TABLE I

Common Multi-Camera Datasets Used in Tracking Evaluation (V=Views, L=Length(min) - Note the Short Length of Most Sequences and Their Specific Content)

Dataset	V	L	fps	Content and/or application
ETISEO [13]	4	18	12.5	Surveillance in an airport apron
PETS2001 [14]	2	3	30	Surveillance of people and cars
PETS2009 [15]	4	2	7	Surveillance in an University Campus
APIDIS [16]	7	1	25	Basketball game (Sports analysis)
UvA-T [5]	3	7.1	20	Surveillance in train platform
UvA-H [5]	4	7.6	20	People walking (Surveillance)
EPFL-L [17]	4	2	25	People walking around in a lab-room
EPFL-C [17]	3	5.2	25	Surveillance inn a University Campus
EPFL-P [17]	4	1.7	25	Surveillance in a passageway
ISSIA [18]	6	2	25	Football game (Sports analysis)

Existing multi-view trackers are usually evaluated on relatively simple content such as people walking or standing without the presence of furniture and — for practical reasons — on short sequences, e.g. [5]. For real-life applications there is a considerable risk that existing solutions are over-tuned to specific datasets. Therefore, testing on more and larger video datasets is essential to explore when and how often trackers fail under diverse conditions. For single camera computer vision, ground truth is abundant, given the recent development of semi-automatic tools which exploit Amazon Mechanical Turk (MTurk) [6], e.g. VATIC [7]. In comparison, for multi-camera video analysis, ground truth is far more scarce. Table I lists the most popular multi-camera datasets (with overlapping views) which have been used to evaluate trackers. Most of these datasets were manually annotated using tools like ViPER [8]. This available ground truth is very useful, but it does not always represent the desired experimental conditions for new research, which may require different camera resolutions or frame rates, different numbers or types of cameras with different viewpoints, different environmental conditions, etc. What is needed are more efficient procedures to generate annotations, or a substitute for it, in *new* datasets.

Generation of references for evaluation is of vital importance for assessing and improving the performance of detectors and trackers. Comprehensive ground truth databases are mostly created using labour-intensive manual annotation of videos, which usually restricts the database to short sequences [5]; one approach is to rely on crowd-sourcing, as has been done for detailed annotation of persons in single images [9] and for delineating bounding boxes in videos [7]. However, even this approach is still labour intensive and requires considerable training and validation to ensure quality of the results.

1057-7149 © 2016 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

For creating ground truth in multi-camera video only manual annotators [10], [11] have been reported so far. One solution would be to apply a single camera system like VATIC for every view independently. However, even such semi-automatic approaches are very labour intensive. For instance, Vondrick *et al.* [7] report that annotating 24 hours of video took about 8 man months of manual annotation time; the reported application was car tracking of possibly many cars in a single view. Moreover, this strategy of employing semi-automatic systems for single view annotation leaves open the question of how to fuse the results from multiple cameras. In any case, we are not aware of any method for generating reference tracking results for multi-camera tracking in a semi-automatic and efficient fashion.

In this paper we therefore propose a procedure for semi-automatic annotation of people tracking in long multi-camera sequences, and provide a new annotated video database where a person performs different activities involving different poses. The methodology aims to produce *reference tracking data* with minimal human effort and with an estimation of precision under given values of desired position accuracy, i.e. the fraction of the resulting annotations with a positional error larger than a specified threshold. With a given amount of human effort, our proposed approach allows producing reference data for much *longer video sequences than is currently possible.* This allows (future) trackers to be evaluated on a much richer test data set, encompassing a wider variety of environmental conditions, with more diverse camera setups, scenes and person appearances.

Specifically, we calculate a position consensus between the output of a so called oracle and many multi-camera trackers. The oracle is a person detector which can be tuned so as to produce highly reliable results, albeit only on a small fraction of video frames. The trackers (*imperfect* oracles [12]) on the other hand produce an output for every video frame but are usually less reliable. In order to fuse the results produced by the oracle and the trackers, we propose a probabilistic framework which is robust to outliers (large tracking errors) and produces a consensus tracking result for all frames. The *consensus tracker* also outputs a *reliability measure* for each frame, which reflects the level of agreement between trackers and the oracle.

In this paper we demonstrate and evaluate the approach on a dataset of 6 hours, taken from 4 views, with furniture present, persons performing different activities, involving different poses and under different light conditions. The consensus tracking result is based on 6 multi-camera trackers, using publicly available software and our own implementations. We quantify the manual labour involved in the proposed methodology by the fraction of video frames that has to be visually inspected and the time that this operation takes.

To evaluate the performance of our methodology, we inspected the consensus tracking results for the full dataset frame-by-frame, correcting the generated annotations where needed. For solely the automatic part, we found that for the whole dataset, considering an accuracy of 60cm, 97.8% of the dataset (420000 frames) was correctly annotated. Using the proposed semi-automatic method, we show that it is possible to isolate most of the wrong annotations and to obtain up to 99% correct annotations. We also present additional results for several performance specifications.

To the best of our knowledge, the proposed probabilistic framework for automatic annotation of person position, using the consensus from many trackers and an oracle, is the first of its kind. Currently our approach is limited to single target tracking. However, we suggest an extension of our method for the multi-target case, which although not fully developed, it shows promising results in common benchmark datasets. In summary, the main contributions of this paper are:

- A methodology for generating reference tracking data in long multi-camera videos, based on the consensus of a detector and several trackers. For multi-camera annotation, our methodology is the first to estimate the reliability of annotations, and to offer the possibility of balance accuracy and human-effort in the final annotation result.
- A novel probabilistic framework to learn the error models of trackers, and how to apply them to estimate target position. Previous methods did not model the tracking error statistics of multiple trackers to increase the reliability of the estimated position.
- The evaluation of the accuracy and reliability of the proposed methodology on a 6 hour dataset, by a comprehensive visual inspection. Scalability of a semi-automatic methodology for annotation in multi-camera data sets of such length is addressed for the first time.
- We make the new calibrated multi-camera dataset available to the community, and provide the annotation results.[1]

The rest of the paper is organized as follows. In Sect. II we summarize related work on generating reference data for tracker evaluation. Sect. III introduces the concepts of our method. It presents the consensus tracker and the procedure for manual inspection. Sect. IV discusses some implementation details. Sect. V presents a long-duration multi-camera tracking experiment for which we generate reference data using the proposed approach. We evaluate the performance of the proposed methodology in Sect. VI. We present an extension and preliminary results for the multi-target case in Sect. VII. We conclude with a discussion of the approach and its limitations in Sect. VIII and future work in Sect. IX.

II. Related Work

In this section we present related work on the evaluation of tracking results. For methods that use position references, we limit our discussion to those where annotations are generated directly on video sequences, either manually or assisted by computer algorithms. Although we are aware of existing methods that generate position references using external sensors and different modalities (e.g. infrared [19], MoCap [20], inertial [21]), we do not include them here as such sensors are not always available or their use is limited to controlled environments.

[1] All the supplemental electronic material (code, results, datasets, videos) necessary to illustrate the concepts described in this paper can be found by accessing: http://telin.ugent.be/~jonino/tip2016/.

Recent work on the evaluation of visual trackers focuses on defining objective metrics for accuracy, failure and robustness [22]–[28] and using these methods to estimate the performance of existing trackers [22], [29]–[31]. This evaluation depends crucially on sufficient ground truth or reference positional data being available and that reference data being of sufficient quality, as demonstrated in the experimental survey in [22]. For the purpose of generating such reference data, several manual annotation tools have been developed for videos from single and multiple calibrated views [8], [10], [11], [32]–[35]. However, these methods do not scale to long video sequences, as manual input is required in every frame. In [36], the editors of a recent special issue on ground truth collection discuss the importance of building efficient semi-automatic tools to generate ground truth for multi-media applications and computer vision research in general. In the following we limit our discussion to tools for positional annotation at the level of image bounding boxes and point references in world coordinates. Thus, we will not consider tools for pixel-wise label propagation in video, like those presented in [37] and [38].

A. Semi-Automatic Tools for Single Camera Annotation

Editing tools can help to significantly reduce human effort in semi-automatic ground truth generation. One such annotation tool is proposed in [39] and made available as a web-based collaborative platform [40]. Generic contour detectors and a tracker are used to reduce human effort for the annotation of bounding boxes in single camera video. However, their contour detection method works only on targets with stable shape and with boundaries that are easily distinguishable from the background, while in more cluttered scenes segmentation errors may cause the tracker to drift. Thus, although a well segmented target may help the tracker, the human has to manually edit or re-draw wrong contours in complicated scenes, which is a tedious task. This method was evaluated on the problem of annotating fish in a fish tank.

An extensive quantitative evaluation of video annotation with crowd-sourced marketplaces and an annotation tool called VATIC, both validated on MTurk, are presented in [7]. In order to reduce human input, humans are asked to manually annotate targets with bounding boxes at periodically selected keyframes. These bounding boxes are then enforced as positional constraints in a dynamic programming framework, which also takes into account target appearance and a simple motion model. The authors propose an annotation cost measure which takes into account both human effort and computational complexity. However, the analysis in the paper is restricted to variants of one tracker which are compared in a single example. In a related contribution by the same authors [41] active learning is proposed to reduce the annotation time of the VATIC tool: an estimate of annotation uncertainty is used as reference to iteratively propose new keyframes to annotate manually. The VATIC approach is one of the most successful ones reported in literature. However, it is restricted to single-view video, and it has some limitations compared to our approach. Firstly, it is not easy to integrate new trackers into the framework. Secondly, it requires humans to annotate bounding boxes in key frames. We believe our approach is less time consuming as the human only has to accept/reject a fraction of the automated annotations, rather than draw bounding boxes, which takes more time. Note that while the VATIC-MTurk framework combines manual effort and computation time, in our work we focus on reducing the manual effort only because computer use is cheap compared to human labor.

In general, compared to our method, the aforementioned semi-automatic techniques do not use the consensus of multiple sources of target position. Because it is well known that a single algorithm cannot cope with scene variability [22], annotation errors are more frequent and human input is still demanding. Methods that fuse the results of multiple algorithms for the purpose of semi-automatic annotation have been proposed recently. A tool that speeds-up the generation of ground truth is described in [42]. The tool is applied in annotations of faces, and combines different face detectors and a tracker for the automatic part of the method, and the ViPER tool [8] for the human intervention. The fusion of the different algorithms relies on the score given by the external detectors. Compared to our methodology, statistics of the errors are not taken into account to fuse the results of the different algorithms. Furthermore, the tool is designed for single camera sequences and there is no estimation of the necessary effort in the manual part of the method.

B. Related Work on Fusing Tracking Results

The idea of fusing the results of several trackers to increase tracking performance appeared in [12], [43]–[45], and [46]. Specifically, Zhong *et al.* [12] approach single-view object tracking as a weakly supervised learning problem, in which the outputs of several trackers are treated as noisy labels. The probabilistic framework for optimal integration of labellers proposed in [47] is used to jointly infer the object position and the accuracy of each tracker. A heuristic that measures *agreement among trackers* is used for training data selection and to update the trackers models. By sampling not only the state of the target but also the trackers involved in its computation, the method proposed in [43] generates proposals for the best combination of tracker components. The target position is then estimated from the selected combination. In [44], a *disagreement-based approach* calculates the weights of a linear combination of probability maps obtained from several trackers. Such a disagreement-based approach is also presented in [45]. In order to find an optimal target bounding box, a measure of *attraction* in a 4 dimensional space (position and size) between bounding boxes from tracking fusion candidates and the trackers is maximised. A heuristic for *bad* trackers removal is proposed to improve the estimate of the bounding box. Other researchers [46] have proposed a *symbiotic* tracker ensemble framework for learning an optimal combination of results from several trackers. This combination is based on estimates of temporary consistency of individual trackers and correlation of pair-wise trackers. In another form of consensus tracking [48], a factorial hidden Markov model learns both the unknown trajectory of the target and the reliability of each tracker. In this case, the consensus position is computed recursively using

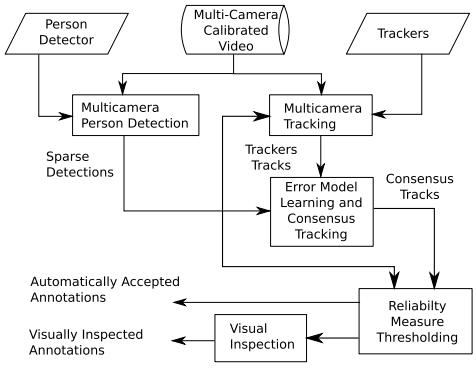


Fig. 1. Flow-chart of the method proposed for semi-automatic annotation.

a particle filter in order to handle multiple hypotheses. The likelihood of each observation (position estimate output by a tracker) is based on the distances between the bounding boxes and the actual target state, where the scale parameter of the chosen distributions is proposed as a reliability measure for each tracker.

The aforementioned methods focus on building a better tracker rather than on presenting a framework for minimal-effort reference data generation by fusing the results of several trackers. However, such *consensus tracking* improves automatic annotations as well, as we will discuss in Sect. V-B. Note that all of these papers deal with single camera tracking only. Moreover, they measure consensus in terms of degree of overlap between bounding boxes, which is only an indirect measure of real-world positional accuracy as the overlap is measured in pixel coordinates which do not have a one-to-one relationship with real-world coordinates. Also, most of these papers report results on small, albeit challenging sequences only, except by the work in [45], which uses the benchmark presented in [29].

C. Semi-Automatic Tools for Multi-Camera Annotation

A semi-automatic tool for annotating players in football video footage proposed in [18] generates proposal tracks using several algorithms: A foreground/background detector extracts blobs for target initialisation; next a simple tracking strategy based on colour information and supported by heuristics, links the blobs over time. A graphical interface is used to edit the bounding boxes produced. Compared to our method, the main disadvantage of this tool is that the user has to browse the video frame-by-frame in order to check/correct the resulting annotations. The authors claim that the annotation tool can also be used for surveillance scenes, even though that type of video content is very different from football. While this is correct in principle, it requires adapting the heuristic rules to specific video content, which is not a trivial task.

III. The Proposed Approach

In this section, we present our proposed method for semi-automatic annotation. Fig. 1 shows the flow diagram of the proposed method. We now summarise this workflow. Our methodology aims at producing consensus tracks by fusing the outputs of several trackers along with a *reliability measure* for each annotation on these tracks; the latter is a measure of the level of agreement between the outputs of the various trackers. This measure is weighted by the positional error model of each tracker, which is learnt automatically in a previous step. By thresholding this measure, we split the consensus annotations according to their reliability. The annotations deemed as reliable are automatically accepted. The annotations deemed as unreliable are subject to a manual procedure, where a human decides on acceptance/rejection of annotated frames. This decision is based on an efficient visual inspection, applied on strategically sampled annotations. To illustrate the concept behind the consensus tracker, we will use the toy example of Figs. 2a-c, showing a person walking between two locations, observed from four views. Note that although the toy example is in itself an easy case for *state-of-the-art* trackers, it explains the basic ideas. These ideas generalize to more complex tracking scenarios, as we will show in Sect. IV. While conceptually simple, the consensus tracking approach presents a number of problems. First of all, trackers need to be initialized before tracking can start and after tracking loss occurs, but not all available trackers implement an initialization method. Also, many trackers are based on similar underlying principles (e.g., foreground background segmentation, colour analysis …) and therefore tend to fail under similar circumstances. Thus, a simple majority-vote consensus or averaging their outputs may end up in wrong position estimates.

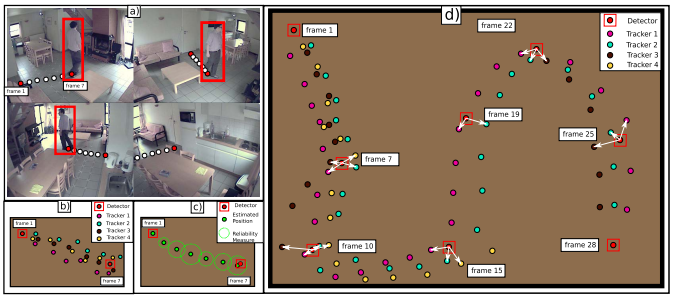


Fig. 2. Illustration of the consensus tracker. (a) The white dots represent the real trajectory of the person. A highly reliable multi-camera person detector produces reference data for only some frames; two successful detections are shown (red dots at frames 1 and 7). (b) Several multi-camera trackers are initialised at frame 1 and track the person independently. (c) Consensus tracking result (green dots) and a reliability measure estimate (dotted green circles: the smaller the circle, the higher the consensus among trackers.). (d) Gathering multi-camera tracking statistics. The trackers are re-initialised at frame 28. In between initialisations, statistics on the position error (white arrows) are gathered by comparing to the output of the reliable person detector (red dot with red square). Note that although ideally a tracker would always output a position estimate, in practice trackers may lose the target (trackers 3 and 4 in this example). Additionally, some trackers may detect target loss and inhibit their output until new evidence is found (tracker 3). Thus, properties like robustness and accuracy are often conflicting between each other and therefore it is desirable to collect tracking statistics of complementary trackers.

Our proposed consensus tracker is based on simple probabilistic models of the tracking errors of individual trackers. One approach would be to ask a human to annotate a number of video frames, and then use this data to compute ground truth positions from which error statistics can be derived. However, we propose a different approach which avoids human labour in this "training" phase. We propose to use an *oracle*, i.e., a method producing position estimates with a high *reliability*

that the postulated position indeed corresponds to a person, rather than to another type of object. This oracle almost never proposes very large "outlier" errors, at the expense of producing position estimates only on a fraction of the dataset. Ideally the oracle should be based on different principles than the trackers so as to have an independent failure mode.

As the oracle, we propose a solution based on a person detector [49] applied independently in each camera view. Such a detector typically performs higher level analysis to distinguish between people and other objects. We tune it to a very low false positive detection rate, at the cost of fewer true positives. The people detector is run on each view separately and outputs a bounding box. When a person is detected in multiple views simultaneously, the position of that person is calculated in world coordinates using triangulation. The oracle solves most of the aforementioned problems: it provides initialisation data for those trackers that require it; when its output is available, it is highly reliable and provides an accurate estimate of the position error of the trackers at this point.

To produce annotation data in between oracle positions, we run multiple multi-camera trackers and fuse their output into a *consensus tracking* result. Fig. 2.d illustrates this approach, which involves estimating the *uncertainty* on the position estimate of each individual tracker and using this uncertainty estimate in a Bayesian fusion approach. To estimate the uncertainty of a tracker, we compare the tracker's output to the oracle position estimates, which are assumed to be highly accurate. Tracking errors are then modelled with distributions, obtained on those frames for which the oracle produces a position estimate. The estimated distributions are then used in the consensus tracker, to properly weigh the contributions of individual trackers in a maximum likelihood approach.

The results will show that the consensus tracker combined with person detection considerably outperforms individual trackers, but is still not sufficiently reliable to classify all of the data automatically. As a result we still only obtain position references for part of the data, but this time the subset of annotated frames is already much larger than what we had based only on the oracle.

In order to increase the amount of annotated data even further, we propose a targeted visual inspection step: we present some of the annotated video frames to a human inspector. The human inspector either accepts those frames as being accurate enough or rejects them as having excessive tracking errors. In the latter case, the human inspector is presented with an alternative hypothesis, which again the human can either accept or reject. In order to make the visual inspection more efficient, the decision made on a single inspected frame is assumed valid for a whole set of video frames surrounding it. The start and end of this *tracklet* are selected based on the consensus of the trackers, as explained later. As we will show in the experimental results, in most cases this procedure renders an appropriate decision. The aim of this human inspection is twofold: firstly it allows us to increase the number of annotated frames significantly, with relatively little effort. Secondly, it allows us to separate the more difficult frames for the tracking task: those whose reliability measure is low and no alternative hypothesis is found.

TABLE II
NOTATIONS

M	Number of trackers.
m	Tracker index.
L	Duration of video segments to be processed by the trackers.
i, j	Frame indexes.
n	View index.
k	target id for the multi-target case.
K	Number of targets in the scene at any frame.
$\mathbf{r}(i)$	Actual (unknown) person position on the ground plane at frame i.
$\mathbf{q}(i)$	Ground plane person position, triangulated from person detector outputs in frame i.
$\mathbf{r}_{m,j}(i)$	Ground plane person position in frame i as estimated by tracker m, after initialisation by the person detector in an earlier frame j.
$\hat{\mathbf{r}}(i)$	Position estimated by the *consensus tracker* at frame i.
$Q(i)^n$	Bounding box in image coordinates produced by a person detector in frame i in view n.
$d_{m,j}(i)$	Position error in frame i of tracker m, after initialisation by the person detector in an earlier frame j.
$R(i)$	Reliability of the consensus tracking result at frame i.
$b(i)$	Binary variable which is set to 1 if the person detector result is available at frame i.
$t_m(i)$	Binary variable which is set to 1 if tracker m is *on target*.
$o_m(i)$	Binary variable which is set to 1 if tracker m produces an output at frame i.
$h(i)$	Binary variable which is set to 1 if a human inspector accepts the annotation at frame i.
λ_R	Reliability threshold for automatic annotations. Frames i with $R(i) \geq \lambda_R$ will be treated as correct without further inspection. The optimal value λ_R^* is computed automatically after setting the annotation performance specifications.
c	precision of the dataset (fraction of frames meeting target accuracy).
$\hat{c}$	Predicted precision using all the position references $\mathbf{q}(i)$.
T_v	Number of tracklets in the human-inspected subset.
$\hat{\tau_v}$	Estimated visual inspection time.
$\#S$	Cardinality of a set S
w_p	Assumed average width of a person in cm.
h_p	Assumed average height of a person in cm.
d_{max}	Maximum position error for a tracker to be considered *on target*.
B	Size of the histogram bins to model the accuracy of the trackers.
G	Area in cm^2 of the ground plane where people is tracked.
$c_{\min}$	Minimum desired precision for the semi-automatic annotations.
$d_{\mathbf{r}}$	Desired position accuracy of the automatic annotations.
l_{max}	Maximum allowed tracklet size in the human-inspected subset.
τ_1	Average visual inspection time of one frame.

In the following subsections, we first present the idea of obtaining a highly reliable detector as the main oracle of person position (III-A). We then discuss the multi-camera tracking setup (III-B), the estimation of the probabilistic error models for the trackers (III-C) and the *consensus tracker* (III-D, III-E, III-F). Finally, we discuss the visual inspection step: how frames are selected (III-G, III-H) and the procedure followed by the human inspector (III-I, III-J).

Table II summarizes the notations used in this paper. While our approach handles multiple video sequences naturally, to simplify the notations we treat all video sequences as a single multi-camera video sequence.

A. Reference Positions From Person Detection

In principle our method can operate with any method for person detection as the oracle. Sect. IV provides more details on the specific choice we made for this paper. The person detector is run in each view *n* independently and produces a bounding box $Q(i)^n$ in image coordinates. It is possible that

for a given frame i a person is detected in some of the views but not in others. In the following we will discard detections which are not accompanied by simultaneous detections in at least one other view. When for a given frame i no person is detected, we will consider $Q(i)^n$ undefined. We extract *reference positions* $\mathbf{q}(i)$ by multi-view triangulation. The image points for this triangulation are taken from the bounding boxes $Q(i)^n$ (as we will explain in Sect. IV-A). We also define a binary variable $b(i)$, which indicates whether or not a reference position $\mathbf{q}(i)$ was computed. The case $b(i) = 0$ occurs when an insufficient number of bounding boxes is available, preventing triangulation.

The reference positions play two important roles: they will be used to initialize trackers and to analyse their errors statistically, so as to build a probabilistic model of these errors.

B. Multi-Camera Tracking

Multiple trackers are initialized with a reference position produced by the people detector and then executed. Fig. 2.d illustrates the procedure on an artificial example with four trackers. Comparing the tracks to the results of person detection, it can be observed that some trackers are more accurate, some are more robust to target loss, or some trackers may detect target loss and inhibit their output until new evidence is found. However we can only extract this information automatically at frames in which a position reference is available. On the other hand, it is possible to estimate the level of agreement between trackers in all frames. We will use this information later to estimate models of tracking performance.

We first need to define initialisation frames for the trackers. These initialisation frames form a subset of the reference positions, selected such that initialisation frames occur every $\approx L$ seconds when possible, as will be explained in Sect. IV-B. We then execute the multi-camera trackers, re-initializing them at each initialisation frame. Let $\mathbf{r}_{m,j}(i)$ be the track produced by tracker $m = 1 \dots M$ of the individual trackers, when initialised from the person detector result in frame j, the index $i \geq j$ being the frame number. Note that some trackers produce an output for each frame, whereas others do not produce an output for some frames, e..g., when failing to detect or loosing a target or when an internally computed reliability is too low. To be able to handle all cases, we define the binary tracker output variable $o_m(i)$, which is 1 iff. tracker m produces an output in frame i. The tracks $\mathbf{r}_{m,j}(i)$ will be compared to the reference positions $\mathbf{q}(i)$ with $i > j$ and $b(i) = 1$, to estimate tracking errors, to build models for these errors and to compute the final consensus tracking result. This procedure will be explained in the following subsections. To not over complicate the notations, we will drop j and i when confusion is not possible.

C. Modelling Tracking Errors

We model tracking error using statistics of tracking loss and tracking accuracy. A tracker m is considered to have lost the target in frame i if $\|\mathbf{r}_m(i) - \mathbf{r}(i)\| \geq d_{max}$, where d_{max} is a specified threshold and $\mathbf{r}(i)$ the unknown true person position. We define the binary random variable $t_m(i)$ with $t_m(i) = 0 \iff \|\mathbf{r}_m(i) - \mathbf{r}(i)\| \geq d_{max}$. In case a tracker does not produce an output we assume by convention that $t_m(i) = 0$. We statistically characterize trackers using the following distributions:

- $P(t_1 \dots t_M)$: The joint probability of a specific *joint tracking loss failure mode* $(t_1 \dots t_M)$. We assume this distribution (as an approximation) to be independent of $\mathbf{r}$, and of i, i.e., to have no spatial or temporal variation. This allows $P(t_1 \dots t_M)$ to be specified using 2^M probability values. For instance, in the case $M = 3$, $P(t_1 = 0, t_2 = 0, t_3 = 1)$ is the probability that trackers 1 and 2 simultaneously loss the target while tracker 3 functions properly.
- $P(t_m = 0 | o_m = 1)$: This is the probability that tracker m's output is wrong when it produces an output. Note that trackers which always produce an output will typically have higher values of $P(t_m = 0 | o_m = 1)$ than trackers which refuse to produce an output in difficult cases.
- $P(o_m = 0)$: Probability that tracker m produces no output.
- $P(\|\mathbf{r}_m - \mathbf{r}\| \; | t_m = 1)$: The probability density of the tracking errors of a given tracker, for the frames in which no tracking loss occurs.

In order to estimate numerical values for these distributions, we rely on the oracle detector, and focus on those frames for which and oracle output is available. Hence, we assume that the frames for which person detection output is available are representative for the whole dataset. Specifically, we compare the output positions $\mathbf{r}_{m,j}(i)$ of each tracker m with the corresponding oracle $\mathbf{q}(i)$, which is assumed to be perfect, i.e., $\mathbf{q}(i) = \mathbf{r}(i)$. This way we obtain a training set of position errors $d_{m,j}(i) \stackrel{\triangle}{=} \|\mathbf{r}_{m,j}(i) - \mathbf{q}(i)\|$. For $P(\|\mathbf{r}_m - \mathbf{r}\| | t_m = 1)$, we compute the normalised histogram of the errors $d_{m,j}(i)$ with $d_{m,j}(i) < d_{max}$ using equally spaced bins of size B in the range $[0, d_{max})$. By counting the number of frames with $d_{m,j}(i) \geq d_{max}$, we determine $P(t_m = 0 | o_m = 1)$. We estimate $P(t_1 \dots t_M)$ by counting the number of times each tracking failure mode $(t_1 \dots t_M)$ occurs. For this estimation we assume that $t_m = 1 \iff \|\mathbf{r}_m(i) - \mathbf{q}(i)\| < d_{max}$. For $P(o_m = 0)$ we calculate the fraction from the frames $i | b(i) = 1$ for which the tracker m does not produce an output.

Note that the models presented do not take into account temporal dependencies in tracker errors, which could be modeled in terms of a conditional distribution $P(\mathbf{r}_m(i) | \mathbf{r}_m(i-1))$ or temporal dependencies in tracker failure mode. Also, a more advanced probabilistic model [50], [51] could also take into account spatially varying probabilities $P(t_1 \dots t_M | \mathbf{r})$. However, estimating such distributions on training data would be both complex and would require too much training data. For this reason we have opted not to use them.

D. The Consensus Tracker

The consensus tracker computes an estimate $\hat{\mathbf{r}}(i)$ of the true person position $\mathbf{r}(i)$ from the position estimates $\mathbf{r}_{m,j}(i)$, $m = 1 \dots M$ of the individual trackers, which in turn have been initialized by the oracle in frame j. The *consensus*

tracker finds the position $\hat{\mathbf{r}}(i)$ that maximises the posterior distribution

$$P(\mathbf{r}, t_1 \dots t_M | \mathbf{r}_1 \dots \mathbf{r}_M, o_1 \dots o_M) \propto P(\mathbf{r}_1 \dots \mathbf{r}_M, o_1 \dots o_M | \mathbf{r}, t_1 \dots t_M) P(t_1 \dots t_M | \mathbf{r}) P(\mathbf{r}).$$

Assuming that the prior $P(\mathbf{r})$ is uninformative and that the probability of successful tracking and the probability of having a tracker output are the same everywhere in the room, we obtain:

$$\hat{\mathbf{r}} = \arg\max_{\mathbf{r}} P(\mathbf{r}_1 \dots \mathbf{r}_M, o_1 \dots o_M | \mathbf{r}, t_1 \dots t_M) P(t_1 \dots t_M).$$

We also assume conditional statistical independence between individual tracker outputs, given a specific failure mode:

$$P(\mathbf{r}_1 \dots \mathbf{r}_M, o_1 \dots o_M | \mathbf{r}, t_1 \dots t_M) \approx \prod_{m=1}^{M} P(\mathbf{r}_m, o_m | \mathbf{r}, t_m).$$

Although this assumption is somewhat weak (trackers may be affected by occlusions in similar ways and therefore their errors may be correlated), it is mitigated by modelling the joint probability of tracking failure $P(t_1 \dots t_M)$. Thus, this assumption still allows for trackers to lose the target in dependent ways, but only assumes that the details of tracking errors (histograms of errors when $t_m = 1$) are statistically independent. We support this assumption experimentally. We have computed the correlation coefficients of the positional errors between pairs of trackers. Most of the absolute values of these correlation coefficients fall below 0.1, except for three pairs of trackers, where they reach values close to 0.2. Using this assumption, we find $\hat{\mathbf{r}}$ that maximizes

$$P(t_1 \dots t_M) \prod_{m=1}^{M} P(\mathbf{r}_m, o_m | \mathbf{r}, t_m) \tag{1}$$

In practice, given inputs $\mathbf{r}_m$ and o_m, the consensus tracker output is computed as follows: for each hypothesized position $\mathbf{r}$, and for all trackers we compute t_m, by comparing $\|\mathbf{r} - \mathbf{r}_m\|$ to the threshold d_{max}. The computed t_m is then input, together with the inputs $\mathbf{r}_m$ and o_m into eq. (1). The consensus tracker output $\hat{\mathbf{r}}$ is then found by optimising eq. (1).

Now we explain how to calculate the factors $P(\mathbf{r}_m, o_m | \mathbf{r}, t_m)$. As a first case, we consider $o_m = 0$, i.e., tracker m does not produce an output. In this case $\mathbf{r}_m$ is undefined and its value is irrelevant. Moreover, by convention t_m is always 0 in this case (see above). Therefore we assume that $P(\mathbf{r}_m, o_m = 0 | \mathbf{r}, t_m = 0) = P(o_m = 0 | t_m = 0)$. Using Bayes' rule, $P(o_m = 0 | t_m = 0)$ becomes

$$\frac{P(o_m = 0)}{P(o_m = 0) + P(t_m = 0 | o_m = 1) P(o_m = 1)}$$

because $P(t_m = 0 | o_m = 0)$ is always 1: if no output, tracker is considered lost. As the second case, we consider $o_m = 1$, i.e,, when tracker m produces an output. In this case, we distinguish between $t_m = 0$ and $t_m = 1$. Assuming that once tracking is lost, all spatial positions are equally likely, we have:

$$P(\mathbf{r}_m, o_m = 1 | \mathbf{r}, t_m = 0) = B / (G - \pi d_{max}^2)$$

where G is the area of the ground plane where people is expected to be tracked. On the other hand, if we assume that the tracking error $\mathbf{r}_m - \mathbf{r}$ is independent of position and if its distribution is radially symmetric, then we have:

$$P(\mathbf{r}_m, o_m = 1 | \mathbf{r}, t_m = 1) = P(\|\mathbf{r}_m - \mathbf{r}\| \ | t_m = 1)$$

Note that all factors in the probabilities and distributions used in the calculation of (1) are estimated in a training phase, with training data using the person detector as a substitute for ground truth reference, as explained in Sect. III-C.

E. Alternative Hypotheses to the Optimal Consensus

For every frame i we obtain not only an estimated optimal value $\hat{\mathbf{r}}(i)$, but also the corresponding optimal tracking failure mode estimate $(\hat{t_1} \dots \hat{t_M})(i)$, which is a deterministic function of the tracker positions $\mathbf{r}_m(i)$ and $\hat{\mathbf{r}}(i)$. During visual inspection, a human inspector may reject $\hat{\mathbf{r}}(i)$. In this case, we generate a *second best alternative hypothesis* for $\mathbf{r}(i)$ by considering other tracker failure modes if possible. This second hypothesis is found by artificially excluding any hypothesized position $\mathbf{r}(i)$ that results in $(t_1 \dots t_M)(i) = (\hat{t_1} \dots \hat{t_M})(i)$, when maximising (1).

F. Automated Annotation Reliability Measure

In this section, we propose a reliability measure to assess the reliability of the position estimates produced by the *consensus tracker*. This measure is very important, since it will be used to classify automatic annotations into subsets called "accepted," (will not be visually inspected) and "candidate for visual inspection". As a reliability measure R, we propose the logarithm of the probability maximised in eq. (1):

$$R = \log P(t_1 \dots t_M) + \sum_{m=1}^{M} \log P(\mathbf{r}_m, o_m | \mathbf{r}, t_m). \tag{2}$$

This measure takes into account the estimated tracking error model and the probability of tracking failure modes. Note that this value is always negative, with 0 being the highest possible value, corresponding to maximal consensus.

G. Selecting Frames for Visual Inspection

Now we will explain how to select frames for visual inspection, based on the resulting reliability measure, and an estimation of the precision of the consensus tracker annotations. This estimation we will be calculated using the people detections as reference, and will allow us to estimate the time a user would spend during the visual inspection step.

By applying a threshold λ_R to the computed reliability measure $R(i)$ for each frame i, we decide if a frame is automatically accepted or not: it is automatically accepted iff. $R(i) \geq \lambda_R$. An optimal threshold $\lambda_R = \lambda_R^*$ will be computed based on user specifications as detailed in Sect. III-H. In the current section, we investigate how the selected threshold affects the number of frames to be visually inspected, the quality of the resulting annotations and the time needed for the inspection. These numbers are estimated and then presented to the user as part of the process for selecting a threshold which optimally meets user requirements. This will become more clear when presenting our case study in Sect. V.

In the following, let S_c be the set of all frames for which a consensus tracker output is available, and let $S_a(\lambda_R)$ and $S_v(\lambda_R)$ be the subsets of *automatically accepted* annotations and annotations to be *visually inspected,* respectively. The dependency on the threshold λ_R is due to the fact that by thresholding R in S_c we obtain S_a and S_v. For the sake of readability, we may omit this dependency in the notations when confusion is not possible. After human visual inspection of the subset $S_v(\lambda_R)$, we obtain the subset S_m of annotations that have been *manually accepted or corrected* by the user (when possible), and a subset S_r of *rejected* annotations. We define the set of semi-automatic *final* annotations $S_f = S_a(\lambda_R) \cup S_m$. We also define the set S_o as the subset of annotations from S_c for which the detector output is available.

Let S be a set of annotations obtained in any way (e.g., automatic, by visual inspection, …). We will use the term *annotation fraction* of a set S for the value $\#S/\#S_c$. We also define the *precision* of a set of annotations as follows:

$$c(S, d_{\mathbf{r}}) = \#\left\{i \in S |\ \|\mathbf{r}(i) - \hat{\mathbf{r}}(i)\| < d_{\mathbf{r}}\right\}/\#S. \tag{3}$$

where $d_{\mathbf{r}}$ is the desired *annotation accuracy* (see Sect. III-H). In words, this precision is the fraction of annotations in a set which has an accuracy of at least $d_{\mathbf{r}}$ centimeters. In the following discussions we will evaluate and predict precision on many different annotation sets, but the notation will make clear which set we mean.

Obviously the precision $c(S, d_{\mathbf{r}})$ is an important quality criterion. However, it *cannot* be computed on the full dataset because of the absence of ground truth. We can estimate the precision $c(S \cap S_o, d_{\mathbf{r}})$ on the frames for which the people detector output is available. In this case we assume that $\mathbf{r}(i) = \mathbf{q}(i)$, i.e., that the people detector produces a perfect position estimate (it is an "oracle"), i.e., we estimate it by evaluating eq. (3), but with $\mathbf{r}(i)$ replaced with $\mathbf{q}(i)$. We call this estimate $\hat{c}(S, d_{\mathbf{r}})$ and apply it to the different annotation sets, except for S_m and S_f, for which it does not make sense to estimate precision before we apply visual inspection. Note that for instance, $\hat{c}_c(S_c, d_{\mathbf{r}})$ is not necessarily a good estimate of $c_c(S_c, d_{\mathbf{r}})$ because the set of frames for which a people detector is available may not be representative for the whole data set. This will become more clear when discussing the results in Sect. VI-B.

After we apply visual inspection, we obtain a set of manually accepted annotations S_m. In the following we will assume that the set S_m has a precision of 100%, i.e., that $c_m(S_m, d_{\mathbf{r}}) = 1$ because we trust that the human inspector does not make any mistakes. We also assume that when finding wrong annotations, a second hypothesis is always correct, thus $S_r = \varnothing$. This assumption implies that at least one of the trackers is always close enough to the real position of the person. We will investigate the validity of these assumption later in this section and evaluate it in Sect. VI-C. Therefore we can compute an estimate $\hat{c}_f(S_f, d_{\mathbf{r}})$ of $c_f(S_f, d_{\mathbf{r}})$ as follows:

$$\begin{aligned}\hat{c}_f(S_f, d_{\mathbf{r}}) &\approx \frac{\hat{c}_a(S_a, d_{\mathbf{r}})\#S_a(\lambda_R) + c_m(S_m, d_{\mathbf{r}})\#S_m}{\#S_c} \\ &\approx \frac{\hat{c}_a(S_a, d_{\mathbf{r}})\#S_a(\lambda_R) + \#S_v(\lambda_R)}{\#S_c}\end{aligned} \tag{4}$$

which depends on the precision $\hat{c}_a(S_a, d_{\mathbf{r}})$ which is estimated on $S_a \cap S_o$ as explained above.

As the next topic in this section, we estimate the time needed for visual inspection which depends on λ_R (which determined which frames are visually inspected). To keep human intervention at a minimum, in the visual inspection step the user only has to inspect *well-selected* frames of $S_v(\lambda_R)$. Some temporally close frames are automatically added, whenever a user accepts a frame as "correct," taking advantage of the temporal continuity in the joint failure mode of trackers. We define an annotation *tracklet* as a sequence of consecutive consensus annotations with the identical computed tracker failure mode. We have experimentally observed that frames within the same tracklet are likely to have the same accuracies, so we assume that they are all either correct or incorrect. This assumption is reasonable in practice provided that tracklets are sufficiently short. For this reason we restrict tracklet duration to at most $l_{max} = 20$ frames. Given the assumption, after sampling an unreliable annotation for visual inspection, we therefore extrapolate the decision of the user to accept or reject the annotation for a single frame to the complete tracklet. The total time needed for visual inspection will therefore be a function of the number of tracklets T_v rather than of $\#S_v(\lambda_R)$. As mentioned earlier in Sect. III-E, when an annotation $\hat{\mathbf{r}}(i)$ is rejected during visual inspection, a second annotation hypothesis is presented to the user. Therefore, the estimate of the time needed for visual inspection is proportional to the number of tracklets within $S_v(\lambda_R)$, plus the number of times a second hypothesis is presented to the user, which depends on $\hat{c}_v(S_v, d_{\mathbf{r}})$. More specifically the visual inspection time is estimated as

$$\hat{\tau_v}(S_v, d_{\mathbf{r}}) = \tau_1 T_v(2 - \hat{c}_v(S_v, d_{\mathbf{r}})), \tag{5}$$

where T_v is the number of annotation tracklets contained within $S_v(\lambda_R)$, and τ_1 is the average time the user needs to inspect one frame. After doing some experimental work on judging annotations, we found that the average time spent on inspecting one frame is $\tau_1 \approx 1sec$.

Finally, we define the order in which annotations are selected for visual inspection. This order is important, because each selected frame deems a whole tracklet as either correctly or incorrectly annotated. In order to reduce the error introduced by the assumption that the precision of the subset S_m is 1, we must try to replace as much bad annotations as possible or discard them. Therefore, it is best to show to the user first the annotation that is more likely to be wrong within a tracklet. It is more likely to find wrong annotations at lower values of reliability measure. Therefore, among the annotations remaining in the set $S_v(\lambda_R)$, we select the frame i with the lowest value of $R(i)$ at every visual inspection iteration.

H. Meeting User-Defined Specifications

To determine which frames will be visually inspected, the user must set *specifications for performance measures*, which will then be translated into an appropriate threshold λ_R, for deciding which frames to automatically accept and to visually inspect. The specifications are: Accuracy of the annotations $d_{\mathbf{r}}$ (the maximum positional error allowed for

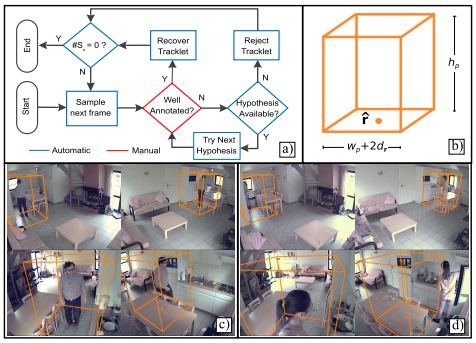


Fig. 3. (a) Visual inspection algorithm. Note that the manual step is limited to decide if an annotation is correct or not. For this decision a cuboid (b) is projected from the annotations $\hat{\mathbf{r}}$, which allows the user to visualise the target within a maximum allowed error. *Accepted* (c) and *Rejected* (d) annotation.

"correct" annotations) and *Minimum precision* $c_{\min}$ (the fraction of final semi-automatic annotations with an actual accuracy no worse than $d_{\mathbf{r}}$). From these numbers, the corresponding optimal value λ_R^* of λ_R can be computed as follows: Eq. (4) allows us to estimate $\hat{c}_f(S_f, d_{\mathbf{r}})$ in terms of the known $d_{\mathbf{r}}$ and the unknown λ_R. Using a numerical root finding procedure we compute the value of λ_R for which $\hat{c}_f(S_f, d_{\mathbf{r}}) \approx c_{\min}$.

I. Human Visual Inspection

The question that remains is how a human evaluator should assess the quality of an annotation. We define a quality criterion based on the desired accuracy $d_{\mathbf{r}}$ and a binary decision $h(i)$ made by the user. Thus, the human evaluator either rejects ($h(i) = 0$) or accepts ($h(i) = 1$) an annotation at frame i according to how well it fits visually the target, within the minimum accuracy $d_{\mathbf{r}}$. Because an annotation $\hat{\mathbf{r}}(i)$ is defined as a 2-dimensional position on the ground plane (point), it would be difficult for a user to visually judge it. Therefore, we augment the annotations by projecting a rectangular cuboid of height h_p (assumed average height of a person) and squared base of side equal to $2 \times d_{\mathbf{r}} + w_p$ centred at $\hat{\mathbf{r}}(i)$, where w_p is the assumed average width of a person and $2 \times d_{\mathbf{r}}$ is a margin for the maximally allowed position error. The user should consider an annotation correct iff. the target is completely enclosed by the projected cuboid in all views, as illustrated in Fig. 3.c.

J. Overview of the Annotation Procedure

In summary, the user must perform the following actions to annotate a new multi-camera video database:

1) Run the automatic stages of our method (detection, tracking, consensus tracking).
2) Graphs showing estimated values of precision, annotation fraction and visual inspection time are displayed for several specification *working points*. (See Sect. V-D).
3) Set the desired specifications: $d_{\mathbf{r}}$ and $c_{\min}$.
4) Perform the visual inspection (See Fig. 3)

IV. Implementation Details

A. Obtaining Reliable People Detections

The proposed annotation procedure is quite flexible. For instance, it can deal with various types of people detectors.

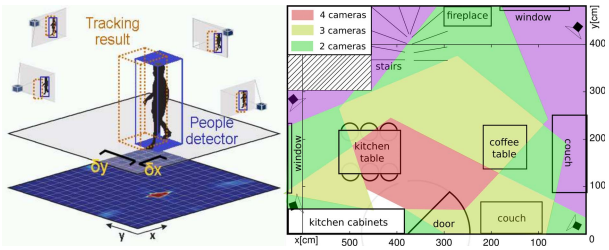


Fig. 4. Multi-Camera setup used to generate the video database and ground plane coverage of the cameras.



Fig. 5. *Poselet* detection in 3 views. Light blue corresponds to the poselet hypothesis with the highest score. The red box is the body detection (details in [49]). Note that with respect to other body parts, the head would be the best reference to reconstruct the real world coordinates of the person position. Therefore, the top-center of the red box (yellow point) is used for triangulation.

In this subsection we discuss the specific choices we used in the case study of sect. V. In this paper we used the *poselet*-based people detector presented in [49], specifically the implementation provided by the authors. We also tested our approach using the people detector presented in [52] and obtained slightly different results.[2]

In order to generate an initial sequence of the most reliable references for person position, we set the "detection score threshold" λ_d following the procedure proposed in [49] using the available ROC curves. We set this parameter for approximately 5% of false positives per image. This resulted in $\lambda_d = 2.8$. Note that we have not re-trained the detector with the images used in our experiments but rather used the existing empirical model made available by the authors. This choice means that there is no need for the user of our proposed annotation procedure to spend additional time on training a people detector. In order to obtain ground plane positions $\mathbf{q}(i)$ from image bounding boxes $Q(i)^n$, we rely on multi-view geometry [53] triangulation. We assume that the top of the head of a person lies on the top-center coordinate of the *poselet* bounding box, as illustrated in Fig. 5. With this point as reference, we estimate the real world coordinates (X, Y, Z) of the top of the head when image *poselet* detections in at least 2 views are available. Note that we keep only the ground plane coordinates (X, Y), as we focus our study on tracking person positions in 2D. In order to ensure reliable references, we only keep ground plane detections where the error in the reprojection of the person head is below a threshold λ_t. We use a value of $\lambda_t = 50 pixels$ for the video sequences used in this paper, which we have found experimentally. For other video sequences a different value can be better, which depends mainly on the multi-camera configuration.

[2]For the sake of completeness, we provide some results obtained with this person detector, specifically the cumulative distribution of tracking errors.

TABLE III
LIST OF TRACKERS USED IN OUR EXPERIMENTS

m	Reference and Description
1	[54] Independent ground plane positions are estimated in each view, using a Kalman filter and detected foreground masks. A fusion center (a second Kalman Filter) integrates the position estimates.
2	[56] Independent image plane positions are estimated in each view, using maximum likelihood estimation from detected texture-based foreground masks. A fusion center estimates the ground plane position that minimise the squared sum of image plane position errors.
3	[17] Multi-camera tracking of objects as a discrete flow optimization problem. Sequential confident maps calculated from foreground masks are represented on a directed graph. Detection linking is solved off-line in batches of a number of frames, using an existing integer optimisation algorithm.
4	[55] Correlation coefficient based template matching applied independently on each view. Fusion integrates the position estimates of all views by computing a weighted average with weight factors related to the correlation coefficients.
5	[57] Visual hull as an approximated 3D model of the target, calculated from detected foreground masks from several views. Occlusions from present furniture are taken into account when calculating the target position.
6	[58] Foreground segmentation and multi-part intensity histogram matching are used for target position measurement in the image plane, within a particle filter framework for each view. Ground plane positions are estimated off-line via multi-view geometry triangulation.

B. Details of Tracking

As explained in Sect. III-B, we aim to process the video sequences in segments of length L to allow for tracking initialisation. The choice of the segments length is rather empirical, as for an optimal value that ensures training/testing data quality one should consider several factors that are unknown beforehand, such as the amount and diversity of motion present. Therefore, we set L to 30 seconds, which results in 410 video segments. In practice however, the value assigned to L is approximate, as we define a window of 5 seconds where the ground plane detection reference with the smallest re-projection error is taken for tracking initialisation. This choice allows us to minimise the risk of wrong initialisation positions. With the initialisation set already defined, we run 6 available multi-camera trackers, five of which have been published in the literature [17], [54]–[57] and a six-th one for which a paper has been submitted [58]. We provide details on the trackers specifications in Table III.

The maximum positional error d_{max}, under which a tracker is considered to be *on target*, is set to 50cm, an experimentally found value which has been reported in *state of the art* multi-camera trackers [5], [17], [59]. To model the accuracy of the trackers, we set the size of the histogram bins to $B = 10cm$.

V. CASE STUDY

As a case study, we apply the proposed annotation procedure, to a data set of 27 multi-view video sequences, containing about 6 hours of content. These sequences were recorded with 4 calibrated cameras in a holiday bungalow with a combined living room and kitchen area, as illustrated in Fig. 4, which shows the ground plane coverage of the cameras. Different sorts of actions (walking, eating, cooking, watching TV, reading, cleaning, using laptop, laying down) involving interaction with moving and static furniture, were recorded, making the video database rich in content and suitable for evaluation of video analysis algorithms.

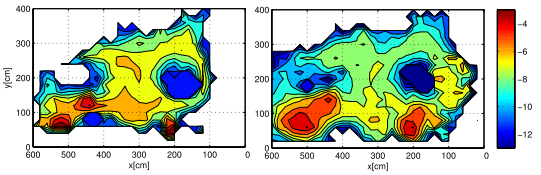
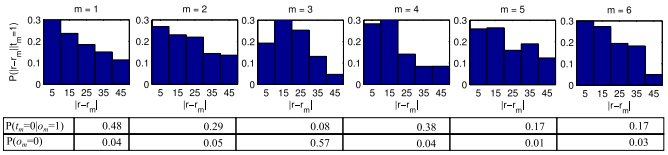


Fig. 6. Spatial distribution (across the room) of detector references (left plot) and positions output by the consensus tracker (right plot). The logarithm of normalised 2-D histograms is calculated, where the color bar represents corresponding values for the different ground plane positions.



$P(t_m=0\|o_m=1)$	0.48	0.29	0.08	0.38	0.17	0.17
$P(o_m=0)$	0.04	0.05	0.57	0.04	0.01	0.03

Fig. 7. For each tracker m: estimated tracking accuracy distribution, resulting target loss ratio, and probability that a tracker produces no output.

Although the case study features only single-target sequences, we included two multi-target sequences from the dataset proposed for the multi-target analysis, which is discussed later in Sect. VII. We now discuss the procedure applied to this case study. Later, in Sect. VI we perform a *meta-analysis* by comparing the annotations to ground truth.

A. People Detection Results

From a total of 420 000 frames, we were able to extract 110 000 frames with reliable person detections (around 26%): 83 000 of those allowed triangulation from 2 simultaneous views, 22000 from 3 views and 5000 from 4 views. In 310 000 frames triangulation was not possible and therefore no position references were obtained. As stated before, detections $\mathbf{q}(i)$ should contain a large variety of person instances in pose and space. Regarding pose, the poselet detector can deal with a wide range of poses [49]. Regarding space, in Fig. 6 we show that the spatial distribution of detections (left plot) is reasonably similar to the distribution of the positions output by the consensus tracker (right plot).

B. Empirical Tracking Error Distributions Per tracker

Fig. 7 shows the distribution of the tracking accuracy $P(\|\mathbf{r}_m - \mathbf{q}\||t_m = 1)$ for the six trackers, the target loss ratio and and the probability that a tracker produces no output. These distributions are used in the computation of the consensus tracker, as explained in Sect. III-D. They show that some trackers are more accurate on average (e.g., $m = 4$ versus $m = 6$) when no tracking loss occurs. The top-left plot in Fig. 8 shows the predicted cumulative distribution of the tracking errors for all individual trackers, the consensus tracker and the average of the trackers outputs. They predict that the proposed consensus tracker outperforms the trackers and the average of the trackers outputs. Note that the consensus tracker benefits from the trade-off between trackers with high accuracy

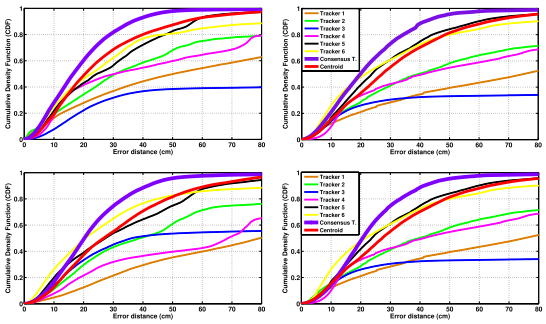


Fig. 8. Top: Cumulative density function of the tracking error (CDF) for the evaluated trackers, the centroid of their outputs and the consensus tracker: predicted (left) and actual (right). The plots at the bottom show the same CDFs, but using the people detector from [52] as reference instead of [49]. In this case, the results show a poorer performance when modelling the tracking errors. However, the overall consensus tracker result is not affected significantly: for the position accuracy, 50% of the annotations have distance error below 20 cm when using [49] and 47% when using [52]. To build these plots, tracking error is considered infinite when a tracker does not produce an output.

and trackers with high recall. For instance, Tracker 3 would not reach more than 40% of the frames, but almost always its positional error would be below $50cm$. On the other hand, Tracker 5 doubles the number of frames with a positional error below $50cm$ ($\approx 80\%$), at the cost of a great deal of frames with higher positional error ($\approx 20\%$). Recall that the results in this subsection were obtained on the subset of annotations S_o, for which people detector references are available. The bottom-left plot in the same Fig. 8 shows the results on the same data set for a variant of our method in which we replace the person detector of [49] by the pedestrian detector of [52]. In general, the results show a poorer performance in modelling the tracking errors. This may be due to the fact that this detector is intended for pedestrian detection, which does not include human poses different than walking and standing.

C. Consensus Tracker Computation and Analysis

The consensus tracker fuses the outcomes of the trackers, taking into account their positional error model. This fusion results in better estimates of the person position, as explained above and illustrated in Fig. 8. The reliability measure R quantifies how trustful this fusion is. In Fig. 9, the top-left plot shows estimated distributions of R, conditioned on the status of the consensus tracker (correct or not), for several values of $d_{\mathbf{r}}$. These distributions predict the range of values R for which is more likely to find wrong annotations, and are estimated on the subset of consensus tracker annotations S_o, for which detector references exist. The bottom-left plot of Fig. 9 illustrates the estimated precision value for automatically accepted annotations as a function of λ_R. The left plot in Fig. 10 shows the fraction of annotations that are automatically accepted for several values of estimated precision, under different values of the user specification $d_{\mathbf{r}}$. We will discuss these predictions again, when analysing the actual results in Sect. VI-B.

D. Analysis of Visual Inspection

Fig. 11 illustrates the predicted visual inspection time and the predicted precision for both the automatic and

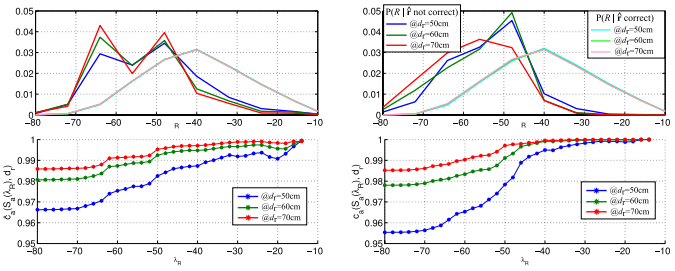


Fig. 9. Top: Distribution of the reliability measure $R(i)$ conditioned on the status of consensus tracker (correct or not according to $d_{\mathbf{r}}$): estimated (left) and actual (right). Bottom: Precision of the automatic annotations (S_a) as a function of the reliability measure threshold λ_R, for several values of the specification $d_{\mathbf{r}}$: estimated $\hat{c}_a(S_a, d_{\mathbf{r}})$ (left) and actual $c_a(S_a, d_{\mathbf{r}})$ (right).

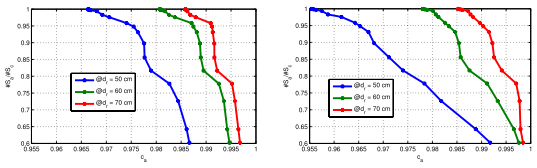


Fig. 10. Annotation Fraction of the Consensus Tracker as a function of Predicted (left) and Actual (right) precision, for several values of $d_{\mathbf{r}}$.

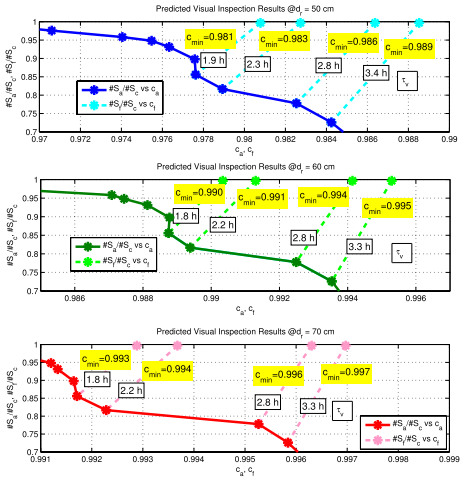


Fig. 11. Predicted results of our semi-automatic annotation method for different specification values. In each row, curves for a given value of $d_{\mathbf{r}}$ are presented. The dashed lines correspond to different desired values of $c_{\min}$ (text labels in yellow boxes). In each case, the dashed line starts at an estimated working point $(\hat{c}_a(S_a(\lambda_R^*), d_{\mathbf{r}}), \#S_a(\lambda_R^*)/\#S_c)$ over the solid line of automatic annotation, and ends at the desired point $(c_{\min}, 1)$. The text labels in white boxes show the predicted visual inspection time.

semi-automatic annotations, for several values of $d_{\mathbf{r}}$ and $c_{\min}$. This figure serves as a guide to the user to select a combination of precision $c_{\min}$ and accuracy $d_{\mathbf{r}}$. The corresponding actual results for these working points will be presented in Sect. VI-C. They will show that in general, between 8%-10% of additional time is spent in visual inspection, regarding the predicted values. This error is due to the overestimation of the precision $\hat{c}_v$, which results in predicting less wrong annotations when a second hypothesis is shown to the user. Details on the errors introduced by the estimation of annotation precision will be discussed in Sect. VI-B.

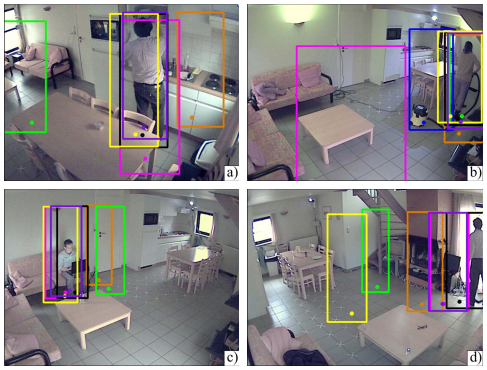


Fig. 12. Cases where the different trackers are challenged by difficult situations in the dataset. a.) The target interacts with moving furniture. b.) The target interacts with objects that move along the target. c.) The target undergoes different poses while performing activities. d.) The scene undergoes illumination changes prompt by the target. The color correspondence for the trackers is the same as in Fig. 8.

VI. Results

In this section, we evaluate the consensus tracker results and visual inspection procedure by comparing the results of the case study to ground truth, obtained by exhaustively inspecting all consensus tracking results, and correcting them where needed. We perform this analysis to illustrate both the strong points and the limitations of the proposed approach. We also compare the manual annotation time of our method against the publicly available VATIC system.

A. Actual Performance of Trackers and Consensus Tracker

The top-right plot in Fig. 8 shows the cumulative density function of the tracking error (CDF) for the evaluated trackers and the consensus tracker, which is computed with respect to the ground truth. Note that, compared to the predicted tracking error CDF (left plot), the prediction is optimistic for trackers 1, 2, 3 and 4 while pessimistic for 5 and 6. The prediction for the consensus tracker is rather optimistic as well. However, it can be observed that the CDF is still similar in both cases, and the general ranking of the trackers remains the same. Thus, in general, the predicted CDF can be used as tool to evaluate trackers, provided a reliable set of position references, and a representative subset of sampled annotations S_o. Note that the results confirm that the proposed consensus tracker outperforms the trackers alone and the average of their outputs. Thus, when fusing the different outputs of the trackers, the proposed modelling of the tracking error results in better position estimates. This is illustrated in Fig. 12.a, b, c, where we show 3 cases where the different trackers are challenged by difficult situations in the dataset. For instance, Fig. 12.b shows a person cleaning the floor, where the vacuum cleaner produces motion disturbance for the trackers. In each case we show that while some trackers fail to locate the target, the consensus tracker retrieves the correct position. The figure also illustrates that, in the other hand, in some difficult cases the human visual inspection needs to be triggered (Fig. 12.d).

B. Meta-Analysis of the Consensus Tracker

In the proposed procedure, the selection of an appropriate working point is an important decision to be made by the user. This decision relies on the estimated precision of the annotations, which at the same time depends on the reliability measure as explained in Sect. III-G. In the following we evaluate the impact of the reliability measure on the quality of the final set of semi-automatic annotations. The top-right plot in Fig. 9 illustrates the actual distribution of R for correct and incorrect annotations. For the analysis, we compare this plot against the top-left plot in the same figure, which shows the corresponding predicted distributions of R. Note that for the distribution of R conditioned on the annotations being wrong ($P(R|\|\mathbf{r}(i) - \hat{\mathbf{r}}(i)\| \geq d_{\mathbf{r}})$), the difference is relatively high, compared to $P(R|\|\mathbf{r}(i) - \hat{\mathbf{r}}(i)\| < d_{\mathbf{r}})$. This behaviour is normal, as far less samples are available to estimate the first one, with respect to the second one. Specifically, for high values of R, the fraction of wrong annotations is over estimated by the samples from S_o. Instead, for low values of R, this fraction is under estimated. This modelling error impacts the estimation $\hat{c}_a(S_a, d_{\mathbf{r}})$ of $c_a(S_a, d_{\mathbf{r}})$, which can be observed by comparing the bottom plots in Fig 9, for which corresponding values of annotation fraction are plot in Fig. 10.

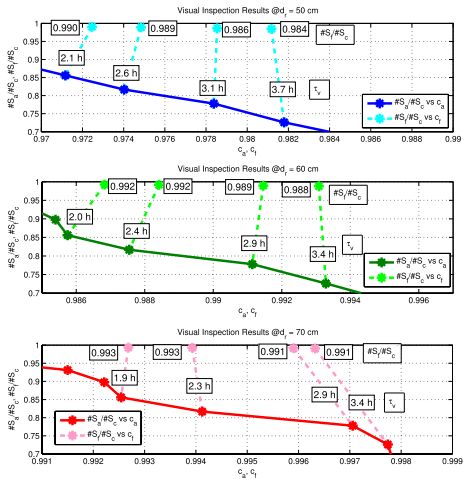


Fig. 13. Actual results of our semi-automatic annotation method for different specification values. In each row, curves for a given value of $d_{\mathbf{r}}$ are presented. The dashed lines correspond to the corresponding different desired values of $c_{\min}$ shown in Fig. 11. In each case, the dashed line starts at the actual working point $(c_a(S_a(\lambda^*_R), d_{\mathbf{r}}), \#S_a(\lambda^*_R)/\#S_c)$ over the solid line of automatic annotation, and ends at the actual point $(c_f, \#S_f/\#S_c)$. The labels in white boxes show the final annotation fraction and the actual visual inspection time.

C. Results of the Visual Inspection

To judge the visual inspection results, we analyse the resulting annotation fraction and precision of the semi-automatic annotations and the time spent during human visual inspection. Fig. 13 illustrates these numbers for several values of position accuracy $d_{\mathbf{r}}$. The plots for results of the automatic annotation (solid lines) and visual inspection (dashed lines) must be compared against the corresponding plots in Fig. 11. In reality the user would of course perform the visual inspection for only one of the working points. For instance, Fig. 11 shows that for an accuracy of 60cm, and a desired precision $c_{\min}$ of 0.991 our method predicts an increase of the annotation fraction from 0.817 (automatic) to 1.0 and (semi-automatic); it also predicts that visual inspection will take 2.2 hours. As illustrated in Fig. 13, the corresponding actual value of precision c_f for this

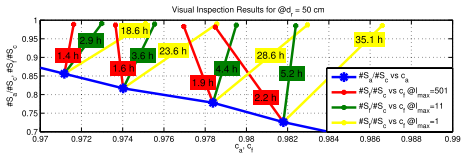


Fig. 14. Results of our semi-automatic annotation method for $d_{\mathbf{r}} = 50cm$ and $c_{\min}$ values given in the corresponding plot in Fig. 11. In this case, we show the visual inspection time for different values of max. tracklet size l_{max}.

working point is 0.988, for a final annotation fraction of 0.992; the actual visual inspection takes 2.4 hours.

In general, the error in the obtained precision c_f respect to the desired precision $c_{\min}$ is due to modelling errors when calculating the working point for visual inspection (as explained in Sect. VI-B) and the error introduced by manually accepted tracklets that contain wrong annotations. Regarding the obtained annotation fraction $\#S_f/\#S_c$, ideally after visual inspection all frames in the dataset are annotated. In practice however, in some inspected frames the user rejects both the optimal and second hypothesis for annotation. In those cases, the corresponding tracklet is discarded and moved to the subset of rejected annotations S_r. Frames that end up in this set are deemed as difficult cases.

These results were obtained for the value of maximum tracklet size $l_{max} = 20 frames$, which we set as a good compromise between visual inspection time and precision of the inspected annotations. We have explored this compromise, as illustrated in Fig. 14. In this figure, precision and visual inspection time are plotted for several values of maximum tracklet size l_{max}, the specification values for $c_{\min}$ given in the corresponding plot Fig. 11 and the specific value $d_{\mathbf{r}} = 50cm$. The yellow lines show the extreme case where every frame in S_v is inspected.

D. Comparison Against Semi-Automatic Annotator VATIC

To compare the manual labour involved in our approach to that of other methods, we also annotated 12 minutes of our multi-camera video with the VATIC system. It took 2 hours of time to annotate 13728 frames. In the proposed semi-automatic method, with $d_{\mathbf{r}} = 50\,cm$, 12800 of these frames were automatically annotated. It took 2 minutes of manual work to inspect the remaining frames. In this set of frames, 6 frames (0.043%) had "wrong" annotations, when compared to the ground truth. One reason for the large speed-up is that with VATIC, semi-automatic annotation has to be carried out on each view separately. The user has to draw bounding boxes (806 in total in this specific case) and check the results of the automatic part of the method using a slider in a video player. In our proposed system, the user judges sampled frames.

VII. The Multiple Targets Case

In this section, we present an extension to the proposed consensus tracker that handles multiple targets. The main problem faced in this case is that of identity switching between targets. This problem could be addressed by modelling the statistics of this specific error mode, in a similar fashion that we deal with the positional error. However, we would need reliable identity references, which are currently difficult to obtain. Instead, we analyse the changes of identity assignment by the different trackers, and detect some of the identity mistakes by relying on the consensus measure of the consensus tracker. Thus, if every tracker is aware of the presence of several targets, the single target approach remains valid but the search space increases according to number of targets present. It is also still possible to trigger manual input in cases where the value R is low and check for the identity switches. In the following subsections we present the specifics of the proposed multi-target extension.

TABLE IV
Possible Tracker-Target Combinations for $M = 4$ and $K = 2$

comb.	$m = 1$	$m = 2$	$m = 3$	$m = 4$
1	$\mathbf{r}_1^1$	$\mathbf{r}_2^1$	$\mathbf{r}_3^1$	$\mathbf{r}_4^1$
2	$\mathbf{r}_1^1$	$\mathbf{r}_2^1$	$\mathbf{r}_3^1$	$\mathbf{r}_4^2$
...	...	...	...	...
16	$\mathbf{r}_1^2$	$\mathbf{r}_2^2$	$\mathbf{r}_3^2$	$\mathbf{r}_4^2$

A. Algorithm

We will extend the notation used before in order to adapt it to the multi-target case. Let $\mathbf{r}_m^k$ be the ground plane position estimated by tracker m of the target k. Let K be the number of targets present at the scene, which we assume to be known by all the M trackers at any moment. In this scenario, we have to calculate K estimates $\hat{\mathbf{r}}$ of the K target positions $\mathbf{r}$. Besides the tracking failures discussed before, we have to account for the possible target identity switch caused mainly by occlusion/disocclusion between targets. Therefore, we have K^M possible *tracker-target combinations*, from which we must find K not mutually exclusive. To illustrate the situation better, let us see the example of Table IV which shows the case for $M = 4$ and $K = 2$. For instance, combinations 1 and 2 are mutually exclusive, because trackers 1,2 and 3 identify the same target ($k = 1$). In the other hand, hypotheses 1 and 16 are plausible together, since in each case the trackers identify a different target.

In order to estimate the K consensus position estimates, we proceed as follows:

1) For every multi-tracker/target combination, we maximse eq. (1) and obtain the corresponding K^M estimates $\hat{\mathbf{r}}$.
2) From this resulting set of possible position estimates, we select the one with the highest value of R. This position estimate corresponds to one of the K targets, and we call it $\hat{\mathbf{r}}^1$. Notice however that at this point we have not yet linked the identity of the target between frames. We will come to that assignment later in Sect. VII-B.
3) For the remaining $K - 1$ targets we repeat step 2, but excluding at every time the mutually exclusive combinations of the preceding estimates. We end up with the set of position estimates $\{\hat{\mathbf{r}}^k\}$, with $k = 1..K$

B. Target Identity Between Frames

For linking target identities between frames, we assume that the true identity of each target is available in at least one frame in the video sequence. In our application this assumption

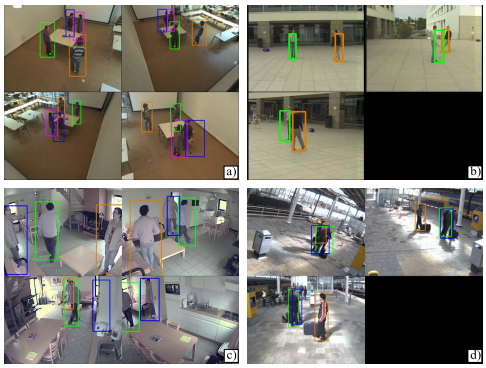


Fig. 15. Some frames from the test sequences for the multi-target case. a.) UGent-L b.) EPFL-C c.) UGent-H d.) UvA-T.

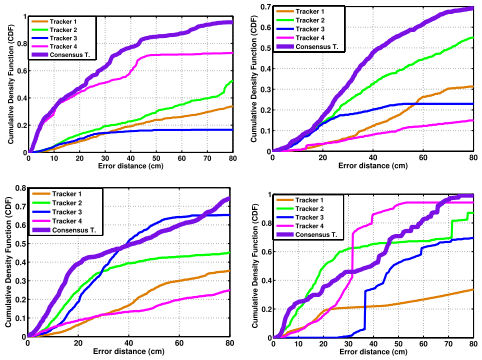


Fig. 16. Cumulative density function of the tracking error for the different trackers and the consensus tracker tested in the multi-target case. Plots for the different video datasets are presented: UGent-L (top-left), EPFL-C (top-right), UGent-H (bottom-left) and UvA-T (bottom-right).

is valid, since for semi-automatic annotation manual user intervention is necessary. We then link target identities between two consecutive frames i and $i+1$ by matching the optimal tracker-target combinations that were found when computing $\{\hat{\mathbf{r}}^k\}$. Let us assume that we know the identity of the targets at frame i. We compare each of the tracker-target combinations of $\{\hat{\mathbf{r}}^k\}(i)$ and $\{\hat{\mathbf{r}}^k\}(i+1)$, and rank the pairings according to the number of configurations tracker/target (m,k) that remain equal between them. We finally select the pairings with higher rank and propagate the target identities from frame i to frame $i+1$.

C. Results

We tested our multi-target extension of the proposed consensus tracker in 4 video datasets, from which two are already publicly available (EPFL-C [17] and UvA-T [5]), one has been tested in a recent publication (UGent-L [54]) and one is new (UGent-H). Fig. 15 shows some sample frames from these datasets.

Although our proposed approach to extend the consensus tracker to the multi-target case performs acceptably (see Fig. 16), there is still room for improvement. The main limitations are the complexity of the algorithm as the number of targets present in the scene increases, and the difficulty of detecting identity switches between targets. Note that, since 2 of the trackers used in the single target case are not capable of handling multiple targets, we excluded them from this scenario.

VIII. Discussion and Conclusion

We proposed a novel method for semi-automated generation of positional annotations in calibrated multi-camera systems, targeted for long video sequences. A key contribution of this paper is a novel procedure to fuse the outcomes of multiple trackers that involves estimating the error statistics of the individual trackers (and their joint failure modes) by comparing their output to a reliable people detector. The *consensus tracker* based on these statistics results in more accurate tracking than averaging the trackers outputs, and the individual output of the trackers. Another novelty is that we proposed a method to estimate the time needed for the manual part of the proposed procedure. We also estimate the precision of the resulting semi-automatically annotated data set, under given values of desired accuracy.

We demonstrated the performance of our method with experiments on a multi-camera video dataset of about 6 hours duration, showing scalability in semi-automatic annotation for long multi-camera sequences for the first time. This leads us to the final contribution of this paper: the annotated data itself, which is available in two versions: the first version results from the proposed procedure; the second from the exhaustive inspection to create ground truth. Since annotated multi-camera data sets of such length are not yet reported in literature, we hope these annotated sequences will be useful to the research community.

IX. Future Research

The proposed approach allows comparing the results of a given tracker to the *consensus* of other trackers. The ultimate goal is to provide a graphical user interface which allows easy visualization of those portions of a data set with large discrepancies between tracking results. In some cases these discrepancies will be due to errors in the reference data. Based on our current internal graphical user interface, it would be easy to allow researchers to indicate segments of video data in which their new tracker outperforms the consensus tracker.

Obviously the proposed methodology has some limitations. For instance, although an extension to deal with multi-target sequences was proposed, we only evaluated the single-person case fully. However, the overall concept applies to multi-person tracking as well. We leave this part as future work.

Regarding the model of positional tracking error, our proposed approach opens the possibility to explore models that take into account the dependency on variables like target pose and location.

Finally, a very important aspect is the time and effort the user has to invest for the manual part of our semi-automatic method of annotation. The visual inspection time can be reduced by optimising the order in which frames with unreliable annotations are shown to the user. In this paper we propose a strategy that prioritizes the seeking of most likely bad annotations; however, other ways of sampling annotations can be explored. Furthermore, by exploring different values

of l_{max}, more efficient working points for visual inspection could be estimated. We leave this part as future work.

REFERENCES

[1] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: A survey," *Comput. Graph. Vis.*, vol. 1, no. 1, pp. 316–323, 2005.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006, Art. no. 13.

[3] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[4] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, Sep. 2013, Art. no. 58.

[5] M. Liem and D. Gavrila, "A comparative study on multi-person tracking using overlapping cameras," in *Proc. 9th Int. Comput. Vis. Syst.*, 2013, pp. 203–212.

[6] Amazon. *Amazon Mechanical Turk*, accessed on Mar. 29, 2016. [Online]. Available: http://www.mturk.com

[7] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 184–204, Jan. 2013, doi: 10.1007/s11263-012-0564-1.

[8] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proc. ICPR*, vol. 4. Sep. 2000, pp. 167–170.

[9] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon mechanical turk," in *Proc. CVPRW*, Jun. 2008, pp. 1–8.

[10] A. Utasi and C. Benedek, "A multi-view annotation tool for people detection evaluation," in *Proc. VIGTA*, 2012, pp. 1–6.

[11] G. Miller *et al.*, "MediaDiver: Viewing and annotating multi-view video," in *Proc. 30th Conf. Human Factors Comput. Syst. Extended Abstracts*, 2011, pp. 1141–1146.

[12] B. Zhong, H. Yao, S. Chen, R. Ji, T.-J. Chin, and H. Wang, "Visual tracking via weakly supervised learning from multiple imperfect oracles," *Pattern Recognit.*, vol. 47, no. 3, pp. 1395–1410, Mar. 2014.

[13] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in *Proc. AVSS*, Sep. 2007, pp. 476–481.

[14] PETS. *Pets2001 Dataset*, accessed on Mar. 29, 2016. [Online]. Available: http://www.cvg.reading.ac.uk/pets2001/pets2001-dataset.html

[15] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proc. PETS-Winter*, Dec. 2009, pp. 1–6.

[16] C. De Vleeschouwer and D. Delannay. (2009). *Basketball Dataset From the European Project Apidis*. [Online]. Available: http://www.apidis.org/Dataset

[17] P. Fua, E. Tu, J. Berclaz, F. Fleuret, and E. Turetken, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[18] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo, "A semi-automatic system for ground truth generation of soccer video sequences," in *Proc. AVSS*, Sep. 2009, pp. 559–564.

[19] S. Sankaranarayanan, F. Bremond, and D. Tax, "Qualitative evaluation of detection and tracking performance," in *Proc. AVSS*, Sep. 2012, pp. 362–367.

[20] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, pp. 4–27, 2010.

[21] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab, "A dataset and evaluation methodology for template-based tracking algorithms," in *Proc. ISMAR*, Oct. 2009, pp. 145–151.

[22] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[23] K. K. Edward, P. D. Matthew, and B. H. Michael, "An information theoretic approach for tracker performance evaluation," in *Proc. ICCV*, Sep./Oct. 2009, pp. 1523–1529.

[24] L. Marcenaro, P. Morerio, and C. S. Regazzoni, "Performance evaluation of multi-camera visual tracking," in *Proc. AVSS*, Sep. 2012, pp. 464–469.

[25] T. Nawaz and A. Cavallaro, "A protocol for evaluating video trackers under real-world conditions," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1354–1361, Apr. 2013.

[26] T. Nawaz, F. Poiesi, and A. Cavallaro, "Measures of effective video tracking," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 376–388, Jan. 2014.

[27] L. Čehovin, M. Kristan, and A. Leonardis, "Is my new tracker really better than yours?" in *Proc. WACV*, Mar. 2014, pp. 540–547.

[28] A. Milan, K. Schindler, and S. Roth, "Challenges of ground truth evaluation of multi-target tracking," in *Proc. CVPRW*, Jun. 2013, pp. 735–742.

[29] H. Wu, A. C. Sankaranarayanan, and R. Chellappa, "Online empirical evaluation of tracking algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1443–1458, Aug. 2010.

[30] M. Kristan *et al.*, "The visual object tracking VOT2013 challenge results," in *Proc. ICCVW*, Dec. 2013, pp. 98–111.

[31] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. ECCV Workshops*, 2014, pp. 191–217.

[32] J. Yuen, B. Russell, C. Liu, and A. Torralba, "LabelMe video: Building a video database with human annotations," in *Proc. ICCV*, Sep./Oct. 2009, pp. 1451–1458.

[33] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008.

[34] A. Dorado, J. Calic, and E. Izquierdo, "A rule-based video annotation system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 622–633, May 2004.

[35] M. Bertozzi, A. Broggi, R. Grisleri, A. Tibaldi, and M. D. Rose, "A tool for vision based pedestrian detection performance evaluation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 784–789.

[36] C. Spampinato, B. J. Boom, and J. He, "MTAP special issue on methods and tools for ground truth collection in multimedia applications," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 409–412, May 2014.

[37] S. Vijayanarasimhan and K. Grauman, "Active frame selection for label propagation in videos," in *Proc. ECCV*, 2012, pp. 496–509.

[38] V. Karasev, A. Ravichandran, and S. Soatto, "Active frame, location, and detector selection for automated and manual video annotation," in *Proc. CVPR*, 2014, pp. 2123–2130.

[39] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato, "A semi-automatic tool for detection and tracking ground truth generation in videos," in *Proc. VIGTA*, 2012, pp. 1–5.

[40] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato, "An innovative Web-based collaborative platform for video annotation," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 413–432, May 2013.

[41] C. Vondrick and D. Ramanan, "Video annotation and tracking with active learning," in *Proc. NIPS*, 2011, pp. 28–36.

[42] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal, "A tool for fast ground truth generation for object detection and tracking from video," in *Proc. ICIP*, Oct. 2014, pp. 368–372.

[43] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. ICCV*, Nov. 2011, pp. 1195–1202.

[44] Q. Li, X. Wang, W. Wang, Y. Jiang, Z.-H. Zhou, and Z. Tu, "Disagreement-based multi-system tracking," in *Proc. ACCV Workshop Detection Tracking Challenging Environ.*, 2012, pp. 320–334.

[45] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *Proc. ECCV*, 2014, pp. 170–185.

[46] Y. Gao, R. Ji, L. Zhang, and A. Hauptmann, "Symbiotic tracker ensemble toward a unified tracking framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1122–1131, Jul. 2014.

[47] J. Whitehill, T.-F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2035–2043.

[48] N. Wang and D.-Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Proc. ICML*, vol. 32. 2014, pp. 1107–1115.

[49] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *Proc. ECCV*, 2010, pp. 168–181.

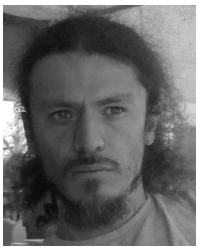
[50] M. Luber, G. D. Tipaldi, and K. O. Arras, "Place-dependent people tracking," *Int. J. Robot. Res.*, vol. 30, no. 3, pp. 280–293, 2011.

[51] J. Saarinen, H. Andreasson, and A. J. Lilienthal, "Independent Markov chain occupancy grid maps for representation of dynamic environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3489–3495.

[52] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *Proc. CVPR*, Jun. 2013, pp. 3666–3673.

[53] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," in *Proc. ACCV*, 2007, pp. 13–34.

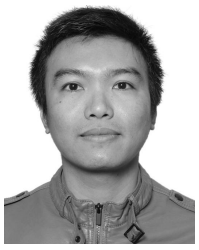
[54] S. Grüenwedel *et al.*, "Low-complexity scalable distributed multicamera tracking of humans," *ACM Trans. Sensor Netw.*, vol. 10, no. 2, Jan. 2014, Art. no. 24.
[55] J. Guan *et al.*, "Template matching based people tracking using a smart camera network," *Proc. SPIE*, vol. 9026, pp. 90260Y-1–90260Y-9, Mar. 2014.
[56] N. B. Bo *et al.*, "Robust multi-camera people tracking using maximum likelihood estimation," in *Proc. ACIVS*, 2013, pp. 584–595.
[57] M. Slembrouck *et al.*, "Self-learning voxel-based multi-camera occlusion maps for 3D reconstruction," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, Jan. 2014, pp. 502–509.
[58] G. Debard, G. Baldewijns, T. Goedemé, T. Tuytelaars, and B. Vanrumste, "Camera-based fall detection using a particle filter," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 6947–6950.
[59] M. C. Liem and D. M. Gavrila, "Joint multi-person detection and tracking from overlapping cameras," *Comput. Vis. Image Understand.*, vol. 128, pp. 36–50, Nov. 2014.



Jorge Niño Castañeda was born in Cunday, Colombia. He received the B.S. degree in electronics engineering and the M.S. degree in control engineering from the University of Ibague, Colombia, in 2005 and 2009, respectively. He is currently pursuing the Ph.D. degree with the Image Processing and Interpretation Research Group, Ghent University, Belgium. His main research interests are in general computer vision and automatic control, and specifically object detection and tracking, and modeling of dynamic systems.



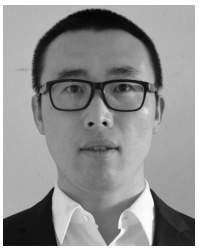
Andrés Frías-Velázquez received the B.S. degree in electronics and telecommunications from the University of Guanajuato, in 2005, and the M.S. degree in information and communication technologies from the Catholic University of Louvain and the Technical University of Catalonia, in 2008. He is currently pursuing the Ph.D. degree with the Image Processing and Interpretation Research Group, Ghent University. His research interests include intelligent surveillance systems, object re-identification, and efficient computer vision algorithms for focal plane processors.



Nyan Bo Bo received the M.Sc. degree from SIIT, Thammasat University, Thailand, in 2008. He is currently pursuing the Ph.D. degree with the Image Processing and Interpretation Research Group, Ghent University, Belgium. He was a Lecturer with the Prince of Songkla University, Thailand, until 2012. His research interests are visual tracking, data fusion in a camera network, and human activity analysis.



Maarten Slembrouck was born in Bruges, Belgium. He received the degree from University College Ghent, and the master's degree in applied engineering, electronics/ICT in 2011. He is currently pursuing the Ph.D. degree with Ghent University, Belgium. He is a Belgian Researcher. He is mainly active in the field of multicamera systems, where he specializes in 3D reconstruction and occlusion handling.



Junzhi Guan received the B.Eng. and M.Eng. degrees from the College of Automation, Harbin Engineering University, Harbin, China, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree with Ghent University, Gent, Belgium. His main research interest is computer vision.



Glen Debard was born in Geel, Belgium. He is currently pursuing the Ph.D. degree with Leuven University, Belgium. He is with MOBILAB, Thomas More Kempen Geel.



Bart Vanrumste received the master's degree in electrical engineering, the master's degree in biomedical engineering, and the Ph.D. degree in engineering from Ghent University, Belgium, in 1994, 1998, and 2001, respectively. He is an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering Technology, KU Leuven, Belgium. His research interests are decision support in healthcare in general and ambient assisted living in particular. His current research activities focus among other on multimodal sensor integration for monitoring of older persons and patients with chronic illnesses at their homes. Since 2013, he has been the Chair of the IEEE Engineering in Medicine and Biology Benelux Chapter.



Tinne Tuytelaars is a Research Professor with KU Leuven, Belgium. Her research interests include image representations, object recognition, and multimodal analysis (combining images and text). She received the Ph.D. degree in applied sciences from KU Leuven, Belgium. She is a member of the Editorial Board of *Computer Vision and Image Understanding* and the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. She was one of the program chairs for ECCV 2014 and is one of the general chairs for CVPR 2016.



Wilfried Philips received the M.Sc. and Ph.D. degrees in electrical engineering from Ghent University, Belgium, in 1989 and 1993, respectively. He is currently a Professor with the IPI Group, TELIN Department, Ghent University, where he is a member of the interuniversitary institute iMinds. He has authored over 300 papers in peer-reviewed journals and international conferences. His main research interests are image and video restoration, image analysis, and lossless and lossy data compression of images and video and processing of multimedia data.