Accurate Positioning via Cross-Modality Training

Savvas Papaioannou Hongkai Wen Zhuoling Xiao
Andrew Markham Niki Trigoni
Department of Computer Science
University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK
e-mail: firstname.lastname@cs.ox.ac.uk

ABSTRACT

In this paper we propose a novel algorithm for tracking people in highly dynamic industrial settings, such as construction sites. We observed both short term and long term changes in the environment; people were allowed to walk in different parts of the site on different days, the field of view of fixed cameras changed over time with the addition of walls, whereas radio and magnetic maps proved unstable with the movement of large structures. To make things worse, the uniforms and helmets that people wear for safety make them very hard to distinguish visually, necessitating the use of additional sensor modalities. In order to address these challenges, we designed a positioning system that uses both anonymous and id-linked sensor measurements and explores the use of cross-modality training to deal with environment dynamics. The system is evaluated in a real construction site and is shown to outperform state of the art multi-target tracking algorithms designed to operate in relatively stable environments.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-Time and Embedded Systems

General Terms

Algorithms, Experimentation, System

Keywords

Wireless Sensor Networks; Tracking

1. INTRODUCTION

To date, the majority of positioning systems have been designed to operate within environments that have long-term stable macro-structure with potential small-scale dynamics. These assumptions allow for stable maps to be produced and gradually aged with the incorporation of minor variations. For example, in an indoor environment, the positions of walls and floors remain constant over time, whereas positions of furniture can change from day to day.

In this work, we tackle the problem of accurate localization in construction sites which are characterized by rapid large-scale changes in structure. For example, Fig. (1) shows the effect of a wall being installed in the middle of one of our tracking experiments. The received signal strength of a worker's smartphone from one of the access points dropped considerably after the installation of the wall, in a matter of minutes. The field of view of the camera also changed, not allowing us to directly visually track the people behind the wall. In addition to these short changes, during our experiments we observed much more significant long term changes (Fig. (2), Fig. (6)); within periods of a few weeks, the scene changed dramatically, staircases or entire floors were added, obfuscating the view to the first floors and creating additional layers where people needed to be tracked.

Currently, there is no system that allows for workers to be tracked reliably and robustly during all phases of construction. As a case in point, consider the challenges in a unified positioning system that works equally as well during deep foundation excavation through to an almost complete

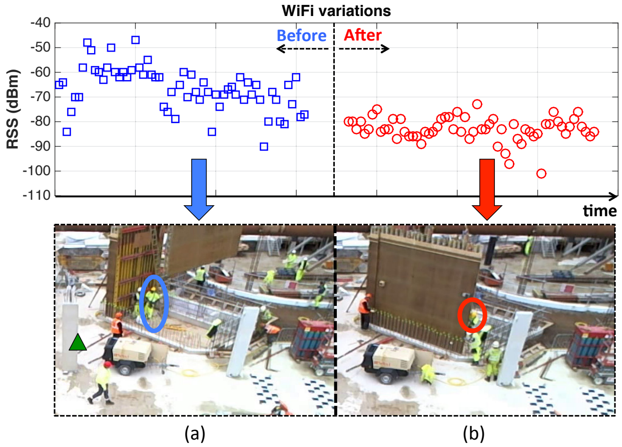

Figure 1: The WiFi signal strength received by the worker in circle is affected by the installation of a new wall (a) Before the installation, there are direct WiFi signals from the access point (shown as triangle) to the worker, (b) The worker is blocked by the new wall, which affects the propagation properties of the WiFi signals as shown in the graph above.

SenSys'15, November 1–4, 2015, Seoul, South Korea.
© 2015 ACM. ISBN 978-1-4503-3631-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2809695.2809712.

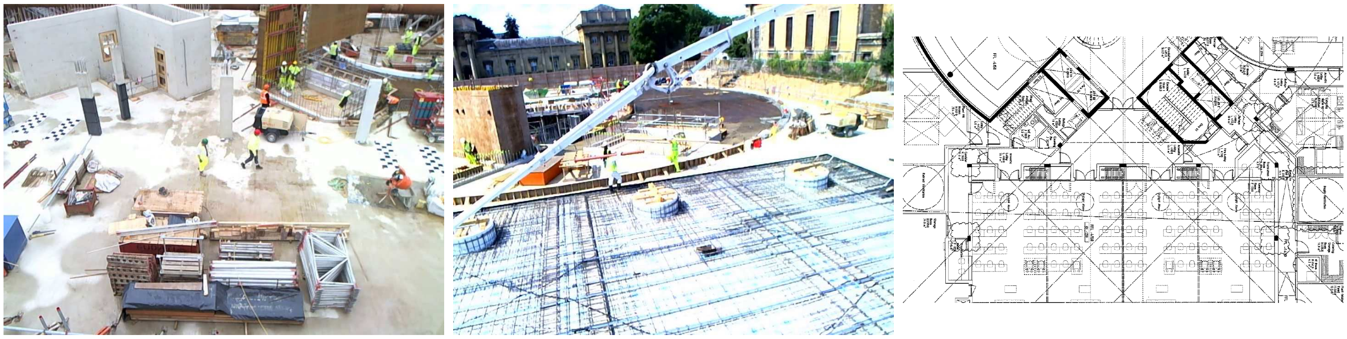

Figure 2: We have conducted tracking experiments in a construction site setting: (a) the construction site on day 1, (b) the construction site on day 36, (c) floor plan of the site. The site changes rapidly from day to day, precluding the use of positioning systems which rely on stable, long-term maps.

multi-storey building. At different points in time, the performance of different techniques alters, with some improving and some degrading. The site itself changes rapidly from day to day, precluding the use of systems which rely on stable, long-term maps (e.g. WiFiSLAM) for positioning.

Our aim is to provide a system that can monitor the location of workers to indicate working hazards (e.g. red and green zones), which can be individually tailored. For example, a steel-worker has the training to operate in areas which might not yet be poured with concrete whilst forming the steel rebar. Conversely, a general construction worker should not venture into regions where steel-work has not been completed. This level of safety requires positioning precision beyond the majority of indoor positioning solutions, with desired sub-meter accuracy.

In addition, for such a positioning system to be used and adopted in the construction industry, it should be low-cost, exploit existing infrastructure and not require manual tuning and calibration. We note that a pervasive feature of construction environments is the use of site-wide closed-circuit television (CCTV) to provide security. CCTV alone is not useful for positioning, as we have to identify what each object in the video sequence actually is. This is especially challenging for low-resolution, grainy footage typically obtained from CCTV, which is unable to accurately distinguish between different people based on facial recognition. Instead however, we note that the task of identifying moving objects from a video sequence is a much easier problem. To tackle the identification problem we use devices carried by the workers which emit radio signals (WiFi/BTLE) and capture inertial measurements to assign identities to each trajectory. This is complicated by the fact that the visual detector is affected by occlusions, changing light conditions and challenges in detection when targets coalesce e.g. if workers are standing together.

We further note that when we have identified a trajectory and linked a set of visual observations with a set of radio and inertial measurements with high probability, we can then use such a trajectory to learn the parameters of all the components in the system. For example, we can learn the radio model, which can help calibrate the system for when the camera is occluded. Individuals also have different gait lengths and whilst a worker is being tracked by the high-resolution vision system, we can exploit this to learn the optimal step-length. This provides us with better measurements when a user steps behind a wall, disappearing from the camera's view. Lastly, we can also learn positions of workers that can actually be observed by a particular camera. This occlusion map is useful for filtering out impossible trajectories. In essence, we are exploiting the fact that different sensing technologies have uncorrelated failure modes to provide a robust, adaptive positioning framework. To summarize, the major contributions of this work are as follows:

1. A positioning framework explicitly designed for rapidly changing environments.
2. A technique for cross-modal sensor parameter learning.
3. A CCTV and smartphone based positioning system.
4. Extensive experiments in a real construction site.

2. PROBLEM DEFINITION

In this paper we tackle the problem of tracking people in environments equipped with one or more stationary calibrated cameras. We assume that people that desire to be tracked carry a mobile device, such as a smartphone or customized worker safety equipment, and move freely in and out of the field of view (FOV). We divide time into short time intervals, and at each time t we receive a number of camera detections of the moving objects denoted as $C_t = \{c_t^1, c_t^2, ..., c_t^j, ...\}, \ 1 \leq j \leq |C_t|$. A camera detection c_t^j represents the bounding box of the j_{th} object generated by a foreground detector as shown in Fig. (4). Note that at time t we could be receiving camera detections not only from people but also from other moving objects (i.e. vehicles); false positive detections are also received due to illumination changes, shadows, etc. In order to reduce the number of false positive detections and concentrate on detecting only people we apply a head detector to the output of a foreground detector.

At time t we also receive a collection of radio measurements $R_t = \{r_t^k\}, \ 1 \leq k \leq K$ where K is the total number of people with mobile devices who wish to be tracked and $r_t^k = [\text{rss}^1, ..., \text{rss}^m]_t^k$ is a vector of received signal strength (RSS) measurements of the k_{th} device from m access points. Additionally, we assume that each mobile device is equipped with an inertial measurement unit (IMU) containing an accelerometer and a magnetometer. This allows us to generate at time t a collection of inertial measurements denoted as

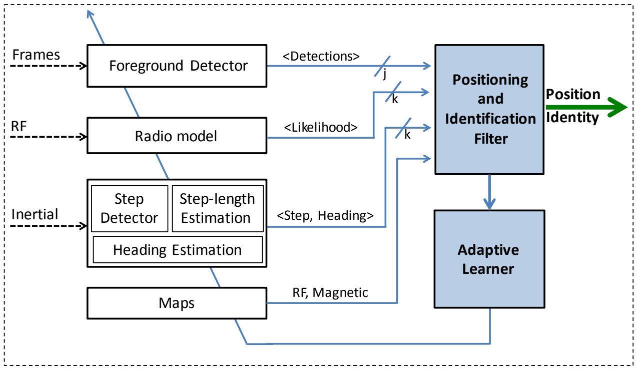

Figure 3: Overview of the proposed system architecture.

$S_t = \{s_t^k\}$ where $s_t^k = [b_t^k, d_t^k, \theta_t^k]$ is a vector that contains the step indicator, step-length and heading of the k_{th} person respectively. Each index k uniquely identifies a person and corresponds to a unique MAC address of the mobile device.

The problem to solve is the following : *Given anonymous camera detections $C_{1:t}$, id-linked radio measurements $R_{1:t}$ and id-linked inertial measurements $S_{1:t}$ estimate the trajectories of all users carrying mobile devices and moving inside the camera FOV.*

3. SYSTEM OVERVIEW

An overview of the proposed system architecture is shown in Fig. (3). The *Positioning and Identification filter* obtains anonymous camera detections, radio and inertial measurements from multiple people and is responsible for solving three problems. Firstly, it establishes the correspondences of camera detections across frames, that is, links together anonymous camera detections that correspond to the same person. Secondly, it finds the mapping between anonymous camera detections and id-linked smartphone (radio and step) measurements. Finally, it identifies and estimates the positions of multiple targets.

The *Adaptive Learner* uses the output of the filter in combination with the input observations, and performs cross-modality training. Specifically, it configures the foreground detector's internal parameters taking into account available motion measurements. In addition, it tunes the step-length estimation method by leveraging reliable camera measurements. Finally, it exploits camera measurements to learn the radio model; radio, magnetic and occlusion maps can also be learned which can be used to further improve the system's accuracy.

The remaining components of the system are existing modules which pre-process raw sensor data and transform them to camera, step and radio measurements.

4. MULTIPLE TARGET TRACKING

In this section we provide a brief overview of previous work on multiple target tracking (MTT). A more detailed description of MTT algorithms can be found in [2].

4.1 Introduction to Multiple Target Tracking

Under the general MTT setup a number of indistinguishable targets are assumed to move freely inside the field of view; they can enter and exit the FOV at random times.

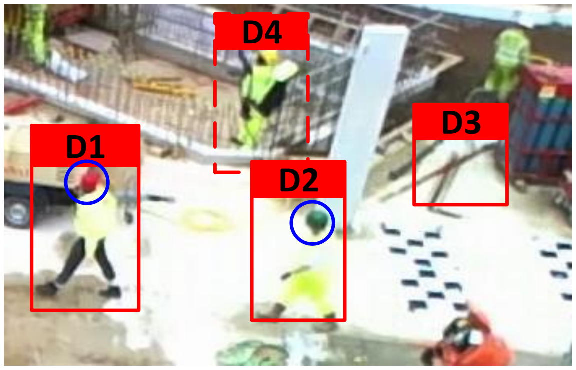

Figure 4: Camera detections generated by our system: a) people are detected correctly by a head detector applied to the output of the foreground detector (D1 and D2), b) Illumination variations create false positive detections (D3). In this case the use of the head detector allows us to discard this measurement, c) not all moving people are detected, the dotted bounding box (D4) shows a case of missing detection.

The system receives sensor data about the position of the targets periodically which are noisy, include false alarm measurements (i.e. background noise or clutter) and occur with some detection probability. Each target follows a sequence of states (e.g. positions) during its lifetime called *track*. The main objective of MTT is to collect sensor data containing multiple potential targets of interest and to then find the tracks of all targets and filter out the false alarm measurements. If the sequence of measurements associated with each target is known (i.e. id-linked measurements) then the MTT reduces to a state estimation problem (e.g. distinct Kalman/particle filters can be used to follow each target). However, when the target-to-measurements association is unknown (for example, anonymous measurements from cameras, radars and sonars are used) the data association problem must be solved in addition to state estimation. Essentially, the data association problem seeks to find which measurements correspond to each target.

MTT algorithms handle both the data-association and the state-estimation problems and they are generally divided into two categories a) *Unique-neighbor data association* and b) *All-neighbor data association*. The former methods allow at most one measurement to be used to update a given track and they usually do not permit a measurement to be used more than once. On the other hand, the all-neighbor data association methods use multiple measurements to update the track estimates. A representative algorithm from the first category is the popular multiple hypothesis tracking (MHT) algorithm [20, 1]. MHT is a deferred decision logic method which maintains multiple track hypotheses. In MHT alternative data association hypotheses are formed whenever there are measurement-to-track ambiguities. The measurement-to-track association decision is postponed until enough measurements are collected that can be used to resolve the association ambiguities. A well known all-neighbor data association method is the joint probabilistic data association (JPDA) filter [8]. JPDA approximates the posterior distribution of the targets as separate Gaussian distributions for each target. At each time step, instead of finding a single best association between measurements and tracks,

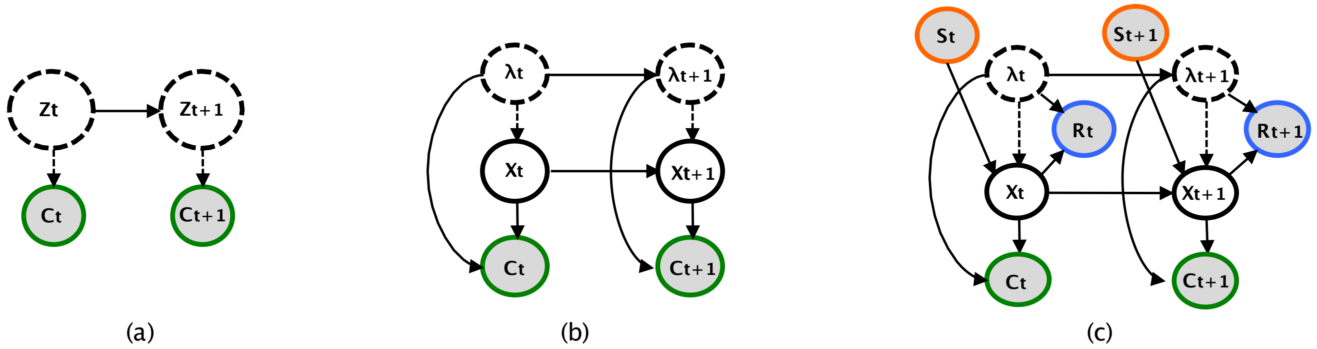

Figure 5: Graphical models for the multiple target tracking problem. Shaded nodes indicate observations and clear nodes indicate hidden variables. Nodes with dashed lines use particle filtering estimation. Dashed arrows indicate that the association between target states and measurements must be recovered before updating a target with a specific measurement. (a) Standard particle filter: uses sampling to estimate the joint posterior distribution of states and data associations ($P(Z_t|C_{1:t})$), (b) In Rao-Blackwellized particle filter (i.e. RBMCDA) the data-association is decoupled from the state estimation. The filter samples only from the data-associations distribution $P(\lambda_t|C_{1:t})$. The distribution of target states conditioned on the association $P(X_t|\lambda_t, C_{1:t})$ is calculated analytically. (c) Proposed approach: id-linked radio (R_t) and inertial (S_t) measurements are incorporated to RBMCDA in addition to the camera detections (C_t). The data association problem is changed compared to (a) and (b) as now we need to recover the association between id-linked measurements and tracks in addition to the anonymous measurements-to-tracks association.

JPDA enumerates all possible associations and computes association probabilities. The state of each target is then updated by every measurement with weights that depend on the association probabilities. The main disadvantage of JPDA is its tendency towards track coalescence for closely spaced targets.

Particle filters (i.e. sequential Monte Carlo methods) have also been used extensively in MTT problems [9] and are typically considered as generalizations of the MHT. Particle filters represent the data association and state posteriors as a discrete set of hypotheses (i.e. particles) and allow for non-linear state-space models.

4.2 Rao-Blackwellized Particle Filtering

In Bayesian inference, the objective is to compute the posterior distribution $P(z_{1:t}|y_{1:t})$ where $z_{1:t}$ are the hidden states at times 1 to t and $y_{1:t}$ are the received measurements up to time t. Suppose we can decompose the hidden state z_t into two parts: a root variable λ_t and a leaf variable x_t as shown below:

$$P(x_{1:t}, \lambda_{1:t}|y_{1:t}) = P(x_{1:t}|\lambda_{1:t}, y_{1:t})P(\lambda_{1:t}|y_{1:t}) \qquad (1)$$

If we can compute the conditional posterior distribution $P(x_{1:t}|\lambda_{1:t}, y_{1:t})$ analytically, then we only need to sample from $P(\lambda_{1:t}|y_{1:t})$ using the particle filter. Thus, the main idea of Rao-Blackwellized particle filtering (RBPF) [6, 5] is to reduce the number of variables that are sampled by evaluating some parts of the filtering equations analytically. This reduction makes RBPF computationally more efficient than the standard particle filter, especially in high dimensional state-spaces.

The Rao-Blackwellized Monte Carlo Data Association filter (RBMCDA) [22, 23] is a sequential Monte Carlo MTT method that uses Rao-Blackwellized particle filtering (RBPF) to estimate the posterior distribution of states and data associations efficiently. More specifically, instead of using a pure particle representation of the joint posterior distribution of states and data associations (see Fig. (5a)), RBMCDA proceeds by decomposing the problem into two parts: a) estimation of the data-association posterior distribution and b) estimation of the posterior distribution of target states. The first part is estimated by particle filtering and the second part is computed analytically using Kalman filtering (Fig. (5b)). The aforementioned decomposition is possible, since in RBMCDA the dynamic and measurement model of the targets are modeled as linear Gaussian conditioned on the data association thus can be handled efficiently by the Kalman filter.

A high level overview of the RBMCDA algorithm is shown in Alg. (1). The algorithm maintains a set of N particles and each particle corresponds to a possible association of anonymous measurements (y_t) to tracks. Each particle maintains for each target its current state x_t (e.g. location) and state uncertainty (i.e. posterior distribution $p(x_t|y_{1:t})$). In the first step (line 4), a Kalman filter is used to predict the next state of a target based on its previous state ($p(x_t|y_{1:t-1})$). Then, the algorithm considers associating each anonymous measurement with each one of the targets in the particle and estimates the probability of each candidate association event (lines 5-6). The association events are modeled with the association indicator λ_t (e.g. ($\lambda_t = 0$) $\implies$ clutter association at time t, ($\lambda_t = j$) $\implies$ target j association at time t, etc). The association probability $\hat{\pi}_j$ for target j is computed from the measurement likelihood $\hat{p}(y_t|\lambda_t)$ and the prior probability of data associations $p(\lambda_t|\lambda_{t-1})$. By sampling the resulting importance distribution, the algorithm selects only one of the candidate associations (line 7) and updates the state of the respective target with the anonymous measurement (line 8). This is repeated for each anonymous measurement (e.g. for each camera detection in the camera frame). The particle's weight is then updated taking into account its previous weight and the probabilities of selected associations (line 9). Once all particles have been updated and their weights normalized (line 11), they are re-sampled based on their normalized weights (line 12). At the end of each iteration, the positions of the targets are estimated as a weighted average (i.e. mixture of Gaussians) across all particles (line 13).

Note that the algorithm above allows us to enforce data association constraints. For instance, we can express that each track is updated by at most one visual measurement, by suitably modeling association priors in line 5.

```
 1: Input: $N$ particles, a measurement vector $y_t$.
 2: Output: $p(x_t, \lambda_t|y_{1:t})$: the joint distribution of target
    states and target-to-measurement associations at time $t$
    given measurements up to time $t$.
 3: for each particle $i \in (1..N)$ do
 4:     For all targets run Kalman filter prediction step.
 5:     Form the importance distribution as:
        For all association events $j$ calculate the unnormalized
        association probabilities:
        $\hat{\pi}_j^{(i)} = \hat{p}(y_t|\lambda_t^{(i)} = j, y_{1:t-1}, \lambda_{1:t-1}^{(i)})p(\lambda_t^{(i)} = j|\lambda_{1:t-1}^{(i)})$
 6:     Normalize the importance distribution.
 7:     Draw new $\lambda_t^{(i)}$ from the importance distribution.
 8:     Update target $\lambda_t^{(i)}$ with $y_t$ using Kalman correction
        step.
 9:     Update particle weight.
10: end for
11: Normalize particle weights.
12: Resample.
13: Approximate $p(x_t, \lambda_t|y_{1:t})$ as:
    $p(x_t, \lambda_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\lambda_t - \lambda_t^{(i)}) \mathcal{N}(x_t|M_t^{(i)}, P_t^{(i)})$
    where $(M_t^{(i)}, P_t^{(i)})$ are the means and covariances of the
    target states of the $i_{\text{th}}$ particle.
```

Algorithm 1: A high-level description of the RBMCDA filter

The existing RBMCDA algorithm is designed to work with anonymous observations. In the next section we point out how we extend it in order to exploit radio and inertial observations that are inherently linked to unique device IDs (i.e. MAC addresses).

5. PROPOSED APPROACH

We are now in a position to describe how we extend the RBMCDA framework to address the identification and tracking problem in a construction site setting. The key difference here is that we introduce id-linked observations in addition to the anonymous camera observations (Fig. (5c)). This impacts a number of steps in the algorithm above as explained in this section.

5.1 State Prediction and Update

As in the original algorithm, each particle uses a set of Kalman filters to track targets; however, in our case, we are not interested in tracking all targets within FOV; we only track people equipped with mobile devices and we continue to do so when they temporarily come out of the FOV. We extend the framework in [22, 23], in order to use id-linked observations in the prediction and correction steps of the Kalman filter. In particular, we use inertial sensor measurements to predict the next state of a person (instead of only relying on the previous state as in line 4). Furthermore, we use WiFi/BTLE and camera measurements to correct the person's state (instead of only anonymous camera measurements as in line 8).

More specifically, the target's dynamics in our system are modeled by the following linear equation:

$$x_t = F_t x_{t-1} + B_t \begin{bmatrix} d_t\ cos(\theta_t) \\ d_t\ sin(\theta_t) \end{bmatrix} + w_t \tag{2}$$

where t denotes the time index, $x_t = [x, y]^{\mathrm{T}}$ is the 2-D state vector of the target's position, F_t is the state transition matrix and the pair (d_t,θ_t) represents the target's step-length and heading respectively. Finally, B_t is a control input indicating whether a step has been taken or not and w_t is the process noise which is assumed to be normally distributed with mean zero and covariance matrix Q (i.e. $w_t \sim \mathcal{N}(0, Q)$). As we already mentioned in Section 2, our objective is to track all people that carry mobile devices. Thus, once we associate a camera measurement to a person ID (i.e. device ID) , Eqn. (2) is used as the predictive distribution of a Kalman filter to model the motion of the identified person using his/her inertial measurements.

Compared with existing techniques [19] that use heuristics to model the human motion, we will show in the evaluation section that the use of inertial measurements in our approach results in more accurate tracking. We should note here that Eqn. (2) is event-based (i.e. based on step events) and events among the different targets are inherently not synchronized. In other words, the steps of different people do not take place at the same time. However, because we need to know the predicted locations of all targets at a specific time, we process Eqn. (2) in a time-based manner. We run the prediction equation for all targets on fixed intervals (i.e. every second) and during that time we find the number of steps taken by each person and we calculate the step-length accordingly. Incomplete steps are handled by accounting only for a percentage of the step-length.

Unlike the original RBMCDA filter that only uses anonymous observations to update the target's state (line 8), in our system a measurement y_t at time t is a vector containing an anonymous location measurement (2D image coordinates transformed to the world plane via a projective transformation) from the camera system and multiple id-linked radio signal strength measurements from people's mobile devices. Thus, the state vector x_t of a target is related to the system measurements y_t according to the following model:

$$y_t = f(x_t) + v_t = \begin{bmatrix} x_t \\ \mathrm{RSS}_1\,(x_t)) \\ \mathrm{RSS}_2\,(x_t)) \\ \vdots \\ \mathrm{RSS}_m\,(x_t)) \end{bmatrix} + v_t \tag{3}$$

where f is a non-linear function that translates the state vector to the measurement domain and v_t is the measurement noise which follows a normal distribution with zero mean and covariance matrix R ($v_t \sim \mathcal{N}(0, R)$). The function RSS_i is given by:

$$\mathrm{RSS}_i(x_t) = P_i - 10 n_i log_{10} \|O_i - x_t\|_2\ ,\ i \in [1..m] \tag{4}$$

where m is the total number of WiFi/BTLE access points and $\mathrm{RSS}_i(x_t)$ is the expected signal strength at location x_t with respect to transmitter O_i. P_i is the received power at

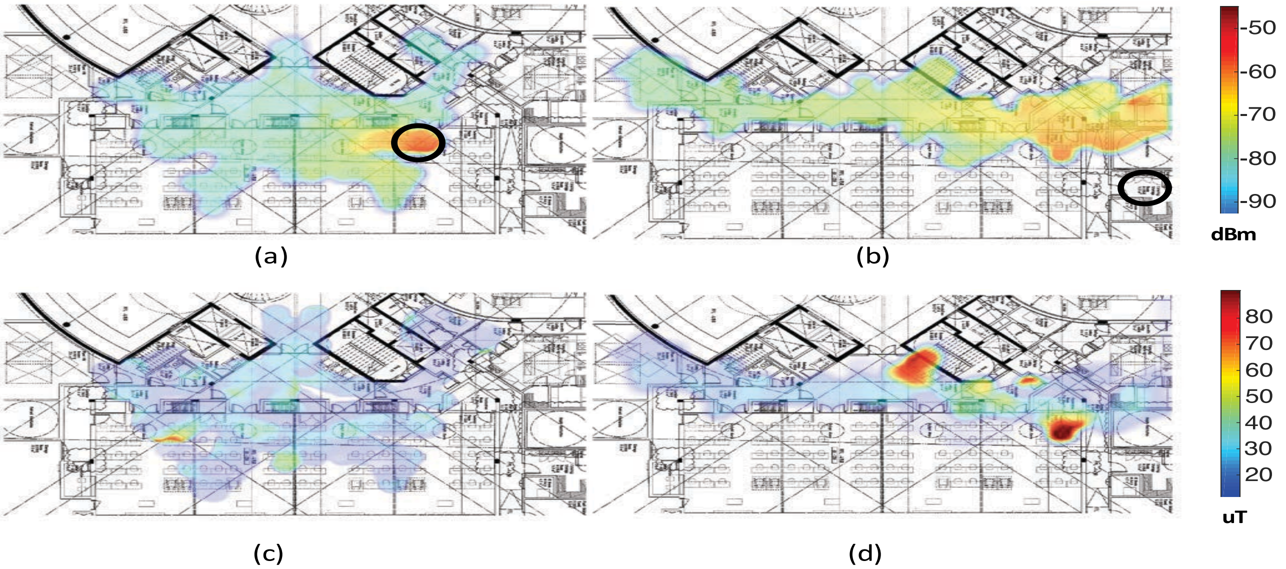

Figure 6: Radio and magnetic maps of the construction site: (a) WiFi map obtained during the first day of the experiment showing the RSS with respect to one access-point (circle denotes the position of the AP), (b) WiFi map obtained 36 days later for the same AP which was moved from its initial location because the ground floor had to been built, (c) magnetic map on day 1, (d) magnetic map on day 36. The environment constantly changes which makes the task of localization and tracking very challenging.

the reference distance of 1 meter and n_i is the path loss exponent. In order to meet the requirements of the RBMCDA filter, i.e. calculate analytically the posterior distribution of the target states with a Kalman filter, Eqn. (3) must be linear Gaussian. The non-linearity of the measurement model in our case is handled via the unscented transformation [10]. Thus, the state estimation can be computed analytically using the unscented Kalman filter (UKF) and each particle contains a bank of UKFs; one filter for each target.

5.2 Tracking and Identification

In this section, we show how we modified the association steps in lines 5-7 to leverage id-linked measurements.

Suppose for instance that at time t we receive camera detections $C_t = \{c_t^j\},\ 1 \leq j \leq |C_t|$ and radio measurements $R_t = \{r_t^k\},\ 1 \leq k \leq K$ where K is the number of people with a mobile device. Each one of the $|C_t|$ anonymous camera detections could be one of the following three types: (a) a person with a device, (b) a person without a device or (c) clutter (e.g false camera detection caused by illumination changes). Our objective is to associate the type (a) camera detections with the correct radio measurements. In order to do that we follow the following procedure. We enumerate all possible combinations $\Omega = |C_t| \times K$ between the camera detections and the id-linked measurements and we create new measurements $y_t^i, i \in [1..\Omega]$ with the following structure:

$$y_t^i = \{c_t^m, r_t^j\},\ m \in [1..|C_t|],\ j \in [1..K] \qquad (5)$$

Now, a measurement y_t^i which contains a correct association will have the following property $\text{RSS}(c_t^m) \approx r_t^j$ for the correct (m, j) pair, where RSS() is the function in Eqn. (4). In other words, if a person is detected by the camera, then his/her radio measurements (i.e. received signal strength) at that location should match the predicted radio measurements at the same location. Camera detections of type (b) and (c) would normally not exhibit the same property. As we have already mentioned the association probability is computed as the product of the measurement likelihood and association prior. The measurement likelihood of associating y_t^i with target j, $\hat{p}(y_t^i|\lambda_t = j)$ is computed as $\hat{p}(y_t^i|\lambda_t = j) = \mathcal{N}(y_t^i; \hat{y}_t, V_t)$ where $\hat{y}_t$ is the expected measurement of target j at the predicted state and V_t is the innovation covariance obtained from the UKF.

Given m simultaneous measurements within a scan the predictive distribution of data associations can be defined as an m_{th} order Markov-chain $p(\lambda_t^m|\lambda_t^{m-1}, ..., \lambda_t^1)$ which allows us to enforce certain association restrictions. In our system this predictive distribution is defined (i.e. assigns zero probability to unwanted events) so that the following conditions are met:

1. A track can be updated with at most one measurement.

2. A measurement can only be used to update at most one track.

3. An already established track (with a specific sensor ID) can only be updated with a measurement of the same sensor ID.

4. Once a camera detection is assigned to a track all other measurements which include the latter camera detection are classified as clutter.

5. A new target is not born if there is an existing target with the same sensor ID as the newborn target. This means that each particle maintains only targets with unique sensor IDs.

Some of the above restrictions can be relaxed depending on the application scenario. For instance, when two people are close to each other they can be detected as one object. In this case the 4th restriction can be relaxed in order to allow two tracks (i.e. two people with different sensor IDs) to be updated with the same camera detection.

To summarize, a particle represents states only for people carrying mobile devices - not for all people in the field of view. Inertial data of each person's device are used to predict their next state. Anonymous camera data are associated with a person's track only if they *agree* with both their inertial and radio data. Finally, we should note here that when at some time-step a particular target does not receive radio measurements then if the target is a new target the identification and creation of a new track is postponed until radio measurements are available. Otherwise, if the target is an existing target, tracking proceeds by only considering the motion model of the target (Eqn. (2)). A high-level work-flow of the proposed technique is shown in Alg. (2).

```
Input: $N$ particles, camera ($C_t$), radio ($R_t$) and inertial ($S_t$) measurements.
Output: $p(x_t, \lambda_t | y_{1:t})$.
Apply Eqn. (5) to $C_t$ and $R_t$ to create $y_t$.
for each measurement $m \in (1..|y_t|)$ do
    for each particle $i \in (1..N)$ do
        For all targets in $i$ run prediction step (Eqn. (2)).
        Form the importance distribution and draw new association event ($\lambda_t^{(i)}$).
        Update target $\lambda_t^{(i)}$ with $m$ using UKF correction step. Update particle weight.
    end for
end for
Normalize particle weights.
Resample.
Approximate $p(x_t, \lambda_t | y_{1:t})$ as in Algorithm 1
```

Algorithm 2: A high-level work-flow of the proposed system.

6. CROSS-MODALITY LEARNING

In this section we will show how our framework is capable of cross-modality learning, i.e. how a subset of sensor modalities is used by the *Adaptive Learner* (Fig. (3)) to train the internal parameters of the system.

6.1 Track Quality Estimation

As we have briefly mentioned in the introduction the output (i.e. track) of our *Positioning and Identification filter* can be used to learn the parameters of various internal components of our system. Once we have identified a track (i.e. we have linked a visual trajectory with radio and inertial measurements), we can use it to learn, for example, the radio propagation model since this track contains all the necessary information (i.e. location-RSS data points) for this purpose. In a similar manner we can learn radio and magnetic maps, train the foreground detector and improve the step-length estimation. All the these will be discussed in more detail later in this section. However, in order to achieve all of the above objectives, we first need to assess the quality of output tracks to make sure that they qualify for the training process. Thus, the goal of the *Track Quality Estimation* phase, is to find candidate tracks which can be used for cross-modality training.

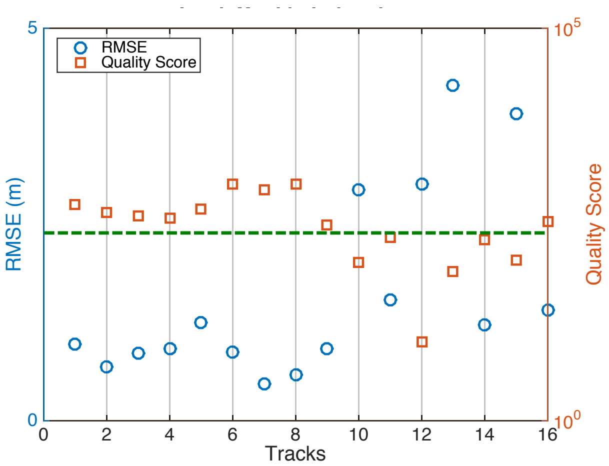

Figure 7: Track quality estimation: The figure shows the quality score of 16 tracks along with their RMSE. Tracks with quality score above the horizontal dotted line are considered qualifying and can be used for cross-modality training

Let us assume that at time-step (or scan) t we receive m measurements $\{y_t^1, y_t^2, ..., y_t^m\}$. In addition y_t^0 is defined for each time-step to be a dummy variable indicating the possibility of a missed detection. Then the incremental quality score of a track j during this time-step is defined as:

$$\Delta L_t^j = \begin{cases} \log\left(\frac{\hat{p}(y_t^i|\lambda_t = j)p_d}{\hat{p}(y_t^i|\lambda_t = 0)}\right) & \text{, if } \exists\ i \in [1..m] \text{ s.t } \lambda_t = j \\ \log\left(1 - p_d\right) & \text{, otherwise} \end{cases}$$

where the quantity $\hat{p}(y_t^i|\lambda_t = j)$ is the likelihood of the measurement assigned to track j. The term $\hat{p}(y_t^i|\lambda_t = 0) = p(clutter)$ is the likelihood of the measurement originating from clutter which has a uniform probability density over the measurement space of volume V (i.e. $p(clutter) = V^{-1}$) and finally p_d is the probability of detection. Then, the cumulative quality score of track j is given by:

$$Q_j = \sum_{t=1}^{T} \Delta L_t^j \tag{6}$$

where T is the total length of the track. As we can see the quality score Q of a track penalizes the non-assignments due to missing detections while favoring the correct measurement-to-track associations. Fig. (7) shows that the quality score is negatively correlated with the root mean square error. Finally, in order to mark a track as a high confidence track that *qualifies* for cross-modal training its quality score is tested against a pre-determined threshold Q_{Th}. If $Q_j \geq Q_{Th}$ then the track is qualified (i.e. *high quality track*) and it can be used for cross-modality training, otherwise the track is rejected (Fig. (7)).

6.2 Foreground Detector Training

The mixture of Gaussians (MoG) [27] foreground detection which is used by our system is one of the most popular approaches for detecting moving targets from a static camera. This approach maintains a statistical representation of the background and can handle multi-modal background models and slow varying illumination changes.

In the original algorithm the history of each pixel is modeled by a mixture of K (typically 3-5) Gaussian distributions with parameters $(w_k, \mu_k, \sigma_k I)$ for the mixture weight, mean and covariance matrix of the k_{th} Gaussian component. In order to find the pixels that belong to the background, the Gaussian distributions are ordered in decreasing order according to the ratio (w_k/σ_k); background pixels exhibit higher weights and lower variances than the foreground moving pixels. The background model is obtained as $B^* = \arg\min_B \left(\sum_{k=1}^{B} w_k > P_b\right)$ where P_b is the prior probability of the background. The remaining $K - B^*$ distributions represent the foreground model.

On the arrival of a new frame each pixel is tested against the Gaussian mixture model and if a match is found the pixel is classified as a background or foreground depending on which Gaussian component it was matched with. If no match is found the pixel is classified as a foreground and it is added to the mixture model by evicting the component with the lowest weight. When a pixel is matched, the weight of that Gaussian component is updated using an exponential weighting scheme with learning rate α as $w_{t+1} = (1-\alpha)w_t + \alpha$, and the weights of all other components are changed to $w_{t+1} = (1-\alpha)w_t$. A similar procedure is used to update the mean and covariance of each component in the mixture.

The learning rate (α) controls the adaptation rate of the algorithm to changes (i.e. illumination changes, speed of incorporating static targets into the background) and is the most critical parameter of the algorithm. Fast learning rates will give greater weight to recent changes and make the algorithm more responsive to sudden changes. However, this can cause the MoG model to become quickly dominated by a single component which affects the algorithm's stability. On the other hand slow learning rates will cause a slower adaptation change which often results in pixel misclassification. Over the years many improvements have been suggested by the research community that allow for automatic initialization and better maintenance of the MoG parameters [4]. More recent techniques [26, 25] address challenges like sudden illumination variations, shadow detection and removal, automatic parameter selection, better execution time, etc .

In this section we propose a novel method for obtaining the optimum learning rate α^* of the foreground detector using the *high-quality* tracks of our filter. Suppose we are given a track $X_{1:T}^j = \{x_1^j, x_2^j, ..., x_T^j\}$ of length T where $x_t^j, t \in [1..T]$ denotes the state of the track at time t. Since, both camera and inertial measurements could have been used to estimate track $X_{1:T}^j$ then its states $x_t^j, t \in [1..T]$ are of two types: type (a) states that have been estimated using camera and inertial measurements and type (b) states that have been estimated only using inertial measurements. A high-quality track ensures that $X_{1:T}^j$ contains the right mixture of type (a) and type (b) states and thus does not deviate significantly from the ground truth trajectory. This is possible, since propagating a track by only using inertial measurements is accurate enough for short periods of time. This key property of the inertial measurements allows us to use a high quality track as if it was the ground truth trajectory to train the learning rate of the foreground detector. In other words the type (b) states of a high quality track tells us that the target is moving to specific locations and the foreground detector does not detect any target at those locations.

The quality score of tracks (Eqn. (6)) can be used to find the optimum learning rate by solving the following optimization problem: *Given a time window $\mathcal{T}$ find a learning rate α^* so that the cumulative quality score (CQS) $\sum_j Q_j$ of all high quality tracks $j \in \mathcal{T}$ is maximized.*

6.3 Optimizing the Step Length Estimation

Similar to the foreground detector training procedure, *high quality* tracks can also be used to learn the step-length model of each person being tracked. More specifically, the step-length of a user can be obtained from the universal model proposed in [21] as:

$$s = h(af_{step} + b) + c \qquad (7)$$

where s is the estimated step-length, h denotes the user's height, f_{step} is the step frequency obtained from the device's accelerometer and (a, b, c) are the model parameters. The model above describes a linear relationship between step-length and step frequency weighted by the user's height.

Since the heights of people that we need to track are not known a priori every time a new track is initialized that contains a sensor ID which has not been recorded before, the step-length estimator uses Eqn. (7) to provide an initial estimate of the target's step-length. At this point the height value is set to the country's average for men of ages between 25 and 34 years old. The parameters (a, b, c) have been precomputed with a training set of 8 people of known heights using foot mounted IMUs.

As the tracking process proceeds high quality tracks are obtained periodically for each target. From these tracks the following IMU data are extracted for each step: a) step frequency, b) step start-time and c) step end-time. The start/end times of each step obtained from the IMU data are then matched to camera detections in order to obtain the position of the target during those times which are essentially the step-lengths measured from the camera system. Thus, for each target we obtain a collection of n calibration points $\{Sv^i, f_{step}^i\}_{i=1}^n$ where Sv^i is the visual step-length of the i_{th} step and f_{step}^i its frequency obtained from the IMU. The calibration set of each target is then used to train a personal step-length model of the form $Sv = w_1 f_{\text{step}} + w_0$ using the least squares fitting. Finally, the step-length estimator can switch to the trained model once the least squares goodness of fit $\left(R^2 = 1 - \frac{\text{residual sum squares}}{\text{total sum squares}}\right)$ exceeds a predefined threshold.

6.4 Radio Model/Maps Learning

High quality tracks are also being used in order to learn the parameters of the radio propagation model which our system uses as explained in Section 5. More specifically, from a high quality track $X_{1:T}^j = \{x_1^j, x_2^j, ..., x_T^j\}$ of length T, the type (a) states are extracted. Let us call a type (a) state as $\tilde{x}_t^j$; this state has been estimated using camera, radio and inertial measurements. Thus a collection of type (a) states $S = \{\tilde{x}_t^j : j \in K, t \in \mathcal{T}\}_n$ of length n where K is the total number of people with smartphones and $\mathcal{T}$ is the running time of our filter, contains n pairs of (location, RSS) measurements. Now, this collection of (location, RSS) points can be used to estimate the parameters of the log-normal radio propagation model [24] given by Eqn. (4) for each access point using least squares fitting.

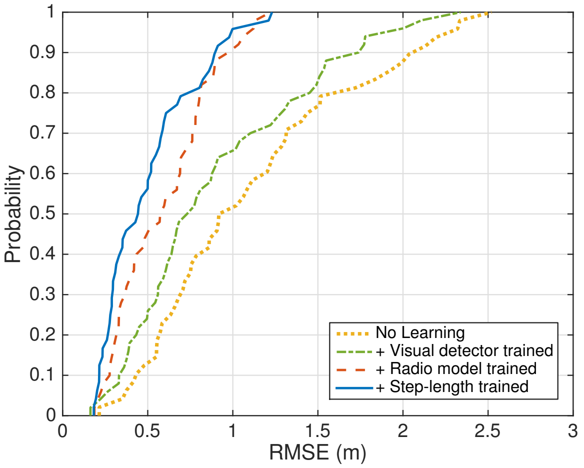

Figure 8: Cumulative distribution function of RMSE for different learning setting.

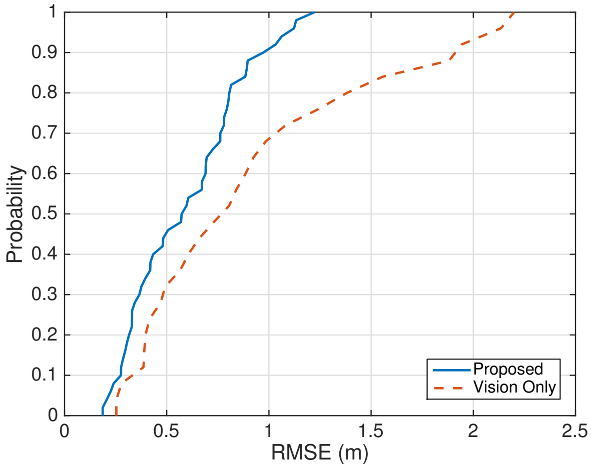

Figure 9: Accuracy comparison of the proposed approach and the original RBMCDA (vision only) algorithm.

At regular intervals we re-estimate the radio model parameters based on the most recent portion of collected data. We should note here that the parameters of the radio model are initialized empirically based on a number of studies for different environments [24].

Additionally, we can follow similar procedure to learn radio, magnetic and occlusion maps. The radio and the magnetic maps can be combined and used for localization in situations where the camera is occluded by an obstacle or they can be used in conjunction with the radio model to improve the system's accuracy. Additionally, the occlusion map, which is derived from the camera detections provides statistics about the environment (i.e. frequent visited areas, inaccessible areas, etc) which our system can use to improve its performance. For instance, suppose that a particular person is not detected by the camera during some time and our filter reverts to IMU tracking; the occlusion map can help us filter out impossible trajectories.

7. SYSTEM EVALUATION

7.1 Experimental Setup

In order to evaluate the performance of the proposed approach we have conducted two real world experiments in a construction site (Fig. (2)). In both experiments we placed two cameras with non-overlapping FOV at approximately 8 meters above the ground facing down. In the first experiment the two cameras were covering an area of approximately 11m × 9m each and in the second experiment an area of 14m × 4m each. The duration of each of the experiments was approximately 45 minutes with the cameras recording video at 30fps with a resolution of 960 × 720 px. We should also mention here that each camera was processed separately (i.e. we do not consider the multi-camera system scenario). The area of the site was outfitted with 12 WiFi and 8 BTLE access points and 5 workers were supplied with smartphone devices. The total number of people in the scene was varying from 3 to 12 as workers were entering and exiting the field of view. The objective of the experiment was to identify and track the workers who were carrying a smartphone device using camera , radio and inertial measurements. The radio measurements were obtained by their smartphones receiving WiFi and BTLE beacons at 1Hz and 10Hz respectively. The inertial measurements (i.e accelerometer and magnetometer) obtained from their smartphones had a sampling rate of 100Hz.

To obtain the ground truth of people's trajectories we followed the same approach proposed in [19]. We supplied all people to be tracked with helmets of different colors and their ground truth trajectories were obtained using a mean-shift tracker [18] to track the colored helmets. We should also note here that our approach does not make use of any appearance or color features for tracking and color features were only used to acquire accurate ground truth.

7.2 Results

The first set of experiments evaluates the tracking accuracy of our system (i.e. how well we can identify and track people with smartphone devices among all people in the FOV). Moreover, we examine what is the effect of cross-modal training on the performance of our system. Our performance metric in this experiment is the root mean square error (RMSE) between the ground-truth and the estimated trajectory. In all the experiments shown here we have used 100 particles. In addition, instead of using line 13 of Alg. (1) to estimate the filtering distribution, in each step the location of each target is estimated using the particle with the highest weight. For this test we used 30 minutes worth of data running our filter on time-windows of one minute (i.e. 1800 frames). Fig. (8) shows the error CDF over this period over all targets for different settings. More specifically, our approach achieves a 90 percentile error of 2.5m when the system is untrained, which improves to 1.8m when the foreground detector is trained. The error decreases further as the parameters of the radio propagation model are learned, achieving a 90 percentile error of 1 meter. Finally, once the optimum step-length of each person is learned the accuracy increases further to approximately 0.8 meters. As we can see the error decreases significantly once both the foreground detector and the radio model are learned. This is expected since our system requires both camera and radio measurements in order to determine the correct measurement to track association and update the target states. In the case of excessive missing camera detections, the trajectory of a tar-

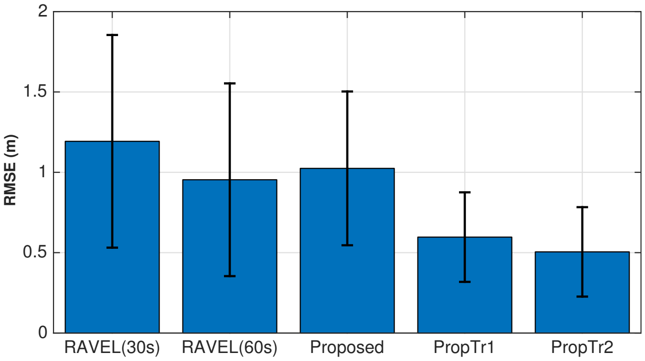

Figure 10: Tracking accuracy between the proposed approach and RAVEL. *Proposed* denotes our approach where the foreground detector and step-length model are not trained. *PropTr1* is our approach after the foreground detector has been trained and further in *PropTr2* the step-length model is also trained. *RAVEL(30s)* and *RAVEL(60s)* is the competing technique evaluated at window sizes of 30 and 60 seconds respectively.

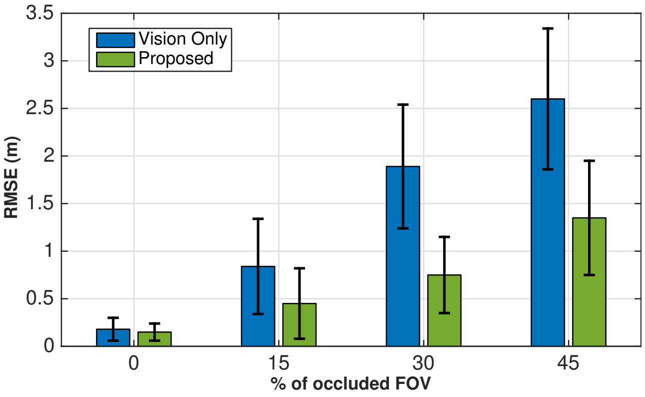

Figure 11: The figure shows the RMSE between the proposed technique and the vision-only tracking for different amounts of occlusion. The use of inertial measurements by the proposed technique improves tracking significantly in noisy scenarios.

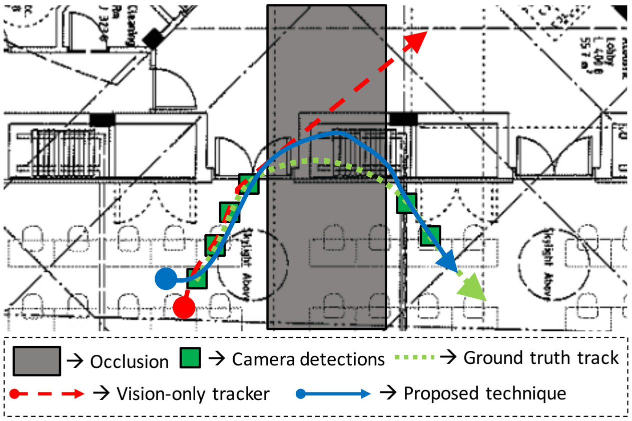

Figure 12: Illustrative example showing the difference between vision-only tracking (red line) and the proposed approach (blue line) in the presence of occlusions (gray area). In cases of prolonged missing camera detections (green squares) the constant velocity model of the vision-only tracker is not sufficient enough to maintain tracking. On the other hand the proposed technique with the aid of inertial measurements is capable of closely following the target despite the presence of long-term occlusions.

get is estimated only by inertial measurements which is the main cause of the low accuracy. On the other hand, if the radio model was not trained, camera detections would not be able to be linked with radio measurements, which would also cause identification and tracking errors. Once the foreground detector and the radio model are trained Fig. (8) does not show any significant improvement after learning the step-length model. This is reasonable since, in this case most of the time the targets are updated with camera observations which are used to correct the predicted by the IMU states. However, from our experiments we have observed that once the camera becomes unavailable, the difference in accuracy between a trained and a universal step-length model is significant.

In our second test we compare the proposed approach with the original RBMCDA algorithm (referred to as vision-only tracker in this section) which uses only visual observations for tracking. In this test we used the same experimental setup as described in the previous paragraph. Both techniques use the same foreground detector settings and in addition the proposed method uses a learned radio model. Fig. (9) shows the error CDF for the two methods. As we can observe the proposed technique achieves a 90 percentile error of 1 meter as opposed to vision-only tracking which has a 90 percentile error of 1.8 meters. The main source of error of the vision-only tracking is due to data association ambiguities which the proposed technique reduces significantly with the help of radio and inertial measurements. Moreover, the proposed technique supports target identification which is not possible when pure visual tracking techniques are used.

The next step is to compare our technique with the recently proposed RAVEL system [19] which is also a multiple hypothesis tracking and identification system. RAVEL which is discussed in more detail in Section 8 exploits the smoothness of motion and radio signal strength data in order to track and identify targets. Unlike our technique, RAVEL is more of a reconstruction technique (i.e. performs off-line tracking) as it requires to observe all measurements over a time window (W) in order to provide the trajectories of each target. We have tested RAVEL using time windows of sizes 30 and 60 seconds over a period of 10 minutes and we have compared it with the proposed online system. Both systems are capable of learning the radio model parameters, thus we performed these tests using the learned radio model for both systems. In Fig. (10) *RAVEL(30s)* and *RAVEL(60s)* shows the accuracy of RAVEL for window sizes of 30 and 60 seconds respectively. *Proposed* denotes the proposed system with learned radio model, *PropTr1* is the proposed system optimized one level further i.e. foreground detector training and *PropTr2* denotes the proposed approach when the step-length model is also learned. Fig. (10) shows that the average error of RAVEL decreases from 1.2m to 0.9m as we increase the window size. Our approach with a trained radio model is slightly worse than RAVEL(60). However,

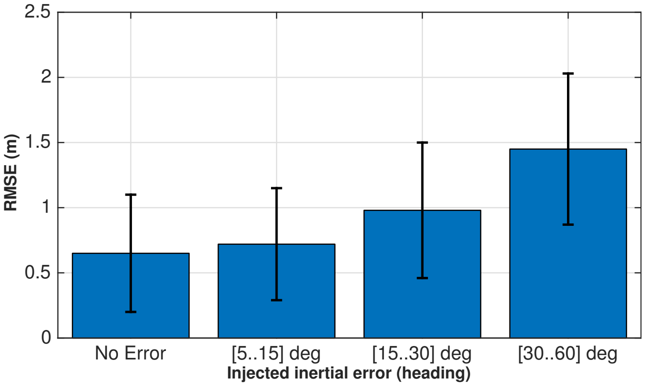

Figure 13: The RMSE of the proposed technique under different amounts of injected heading error.

once our system trains the foreground detector, the average error decreases significantly and continues to decrease as the step-length model is also learned. Unlike our system, RAVEL estimates the trajectory of a target using only visual data thus it becomes easily susceptible to errors due to missing camera detections. Our system without training achieves a similar performance but in real-time.

The final set of experiments aims to demonstrate the robustness of the proposed technique. First we wanted to see how our technique performs on difficult trajectories (i.e. various amounts of occlusions and missing detections). In order to simulate occlusions we remove a specific area of the field of view (FOV) by disabling the camera detections inside that area. More specifically, we generated occlusions at random locations that occupy a rectangular area of specific size inside the FOV. Then we evaluated the accuracy of the proposed approach compared to the vision-only tracker on 50 trajectories of variable length generated from our ground truth data. Fig. (11) shows the RMSE over all trajectories between the proposed system and the vision-only tracker for different configurations of occlusions (i.e. shown as the percentage of occluded FOV). For each configuration we run the test 10 times; each time the occlusion was positioned to a different location. The two methods achieve a comparable performance when there are no occlusions. However, the proposed approach significantly outperforms the vision-only tracking in scenarios with long-term occlusions and large amounts of missing detections. In the presence of long-term occlusions the constant velocity/acceleration motion model utilized by most visual tracking techniques fails and cannot be used to reliably model the inherently complex human motion. On the other hand Fig. (11) shows that the use of inertial measurements by the proposed technique provides a more accurate model of human motion. An illustrative example is shown in Fig. (12).

Additionally in order to study how our approach can cope with variable noise from the inertial sensors we followed a similar procedure as in the previous paragraph and we generated 50 trajectories from our ground truth data. At each time-step and for each trajectory we inject a random bias error to the heading estimator. More specifically we sample a heading error uniformly from a specific range of the form $[a..b]$ *degrees* and we add it to the output of the heading estimator. By doing this we can get an idea of how our approach performs in environments with disturbed magnetic fields. Fig. (13) illustrates the results of this experiment for different amounts of injected noise. As we can see the proposed technique can cope with moderate amounts of inertial noise; achieving a sub-meter accuracy for bias up to 30 degrees.

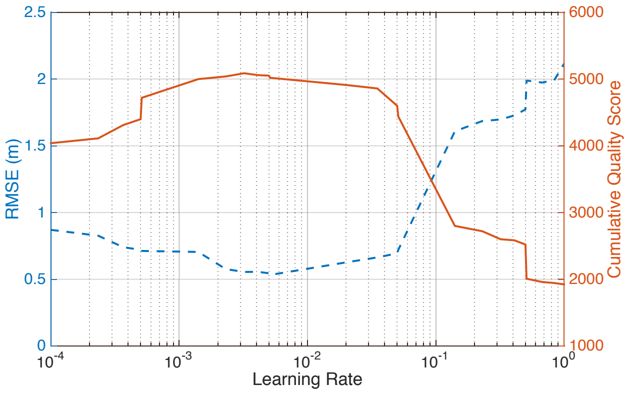

Figure 14: The figure shows the cumulative quality score (CQS) over a period of time as a function of the foreground detector learning rate (α). The optimum learning rate according to RMSE maximizes CQS, thus this metric can be used to train the foreground detector.

Finally, Fig. (14) shows how our approach can find the optimum learning rate (α^*) of the foreground detector by solving the optimization problem discussed in Section 6.2. In the example above we used 5 minutes of data, running the foreground detector for different values of (α) and calculating the cumulative quality score (CQS) for that period. Our intuition is that the optimum learning rate will reduce the number of missing detections, thus increasing the number of high quality tracks as well as their quality score. This is shown in Fig. (14) where the optimum learning rates achieve a high CQS, also evident by the low RMSE.

8. BACKGROUND

A variety of positioning systems have been proposed by the research community over the past ten years. Recent surveys outlining the different techniques and their accuracies can be found in [14, 17]. In this section we will give a brief overview on the most recent positioning systems that make use of radio-, inertial- and visual- sensing (i.e. using a stationary camera) to track multiple people. The positioning systems to be described here can be divided into two categories: a) systems that combine visual and radio measurements and b) those that combine visual and inertial measurements.

Vision+Radio positioning systems: The recent Radio And Vision Enhanced Localization (RAVEL) system [19] fuses anonymous visual detections captured by a stationary camera with WiFi readings to track multiple people moving inside an area with CCTV coverage. The WiFi measurements of each person are used to add context to the trajectories obtained by the camera in order to resolve visual ambiguities (e.g. split/merge paths) and increase the accuracy of visual tracking. RAVEL operates in two phases namely tracklet generation and WiFi-aided tracklet merging. In the first phase visual detections collected over a period of time are used to form unambiguous small trajectories (i.e. tracklets). In the second phase, RAVEL uses the aforementioned tracklets to create tracklet trees for each person (i.e. probable trajectory hypotheses). Then, the WiFi measurements of each person are used to search through the tracklet tree in order to find their most likely trajectory. The most likely trajectory is the one that agrees the most with the WiFi measurements. Unlike our technique, RAVEL performs off-line tracking, i.e. the trajectory of each person is reconstructed after all camera detections and WiFi measurements for a period of time have been observed. In addition, RAVEL does not make use of inertial measurements and thus it is more susceptible to positioning errors due to missing detections (i.e. static people that become part of the background).

In a similar setting the EV-Loc system [29] estimates the position of multiple people using both WiFi and camera measurements. More specifically, EV-Loc estimates the distance of each person from a number of access points first using camera measurements and then using WiFi readings. The Hungarian algorithm [3, 12] is then used to find the best mapping between camera and WiFi measurements. After this optimization problem is solved, the camera and WiFi locations of each person are fused to form a weighted average final location. Unlike our work, EV-Loc concentrates on the problem of finding the best matching between camera and WiFi traces (i.e. the matching process is performed after the visual tracking is completed) and does not provide a general tracking framework that incorporates multiple sensor modalities.

Finally, Mandeljc et al. presented in [16, 15] a fusion scheme that extends the probabilistic occupancy map (POM) [7] with radio measurements. In the original POM framework the area of interest (i.e. the ground plane as viewed by the cameras) is divided into a number of cells forming a grid. Under this framework humans are represented as rectangles, and detections are generated using background subtraction techniques. The algorithm models each cell of the grid as random variable representing the probability of a cell being occupied by a person. Finally the goal of the algorithm is to estimate the probabilities of occupancy for each cell given binary images obtained from a background subtraction process from multiple overlapping cameras. In [16] the POM is extended so that the cell occupancy probabilities are estimated using ultra-wideband (UWB) radio sensors in addition to the cameras. Essentially, the radio measurements are anonymized and they are used to provide a prior occupancy model which is then combined with the camera occupancy model. This additional radio information increases the accuracy and robustness of the algorithm. Later in [15], the POM is extended further so that the anonymous camera detections are augmented with identity information from radio tags. The augmentation of anonymous detections with identity information is done on a frame-by-frame basis where at each time instant the optimal assignment between radio and camera locations is obtained using the Hungarian algorithm. The fusion scheme of [16, 15] was evaluated using only UWB radios which exhibit sub-meter accuracy and there is no indication of how this method will perform with radios of lower accuracy (i.e. WiFi). Although [15] seems similar to our method, it requires multiple cameras with overlapping fields of view, it does not take advantage on inertial measurements, and finally it does not possess the learning capabilities of our method.

Vision+Inertial positioning systems: Instead of using radio measurements for identification the methods in this category use inertial measurements. For instance, the system in [28] fuses motion traces obtained from one stationary camera mounted on the ceiling and facing down with motion information from wearable accelerometer nodes to uniquely identify multiple people in the FOV using their accelerometer node IDs. Background subtraction is used to detect people from the video footage and then their floor-plane acceleration is extracted by double differentiation. The camera acceleration traces are then compared against the overall body acceleration obtained from the accelerometer nodes using the Pearson's correlation coefficient. The acceleration correlation scores among all possible combinations of camera-accelerometer pairs are then used to form an assignment matrix. Finally, the assignment problem is solved using the Hungarian algorithm. The initial algorithm of [28] is extended in [11] to allow for better path disambiguation based on people's acceleration patterns by keeping track of multiple trajectory hypotheses.

More recently the OPTIMUS system [13] uses a similar approach, where inertial measurements from smartphones are used to identify visual trajectories. The algorithm uses histograms of oriented gradients (HOG) descriptors to detect people in a scene covered by one stationary camera and then optical flow is used to group consecutive detections together when there are no ambiguities forming tracklets. A track-level association procedure is then performed to merge the found tracklets and create full trajectories. At the identification stage accelerometer readings from the smartphones are converted into binary vectors that indicate the user's movement and are matched with movement vectors extracted from the visual trajectories using the Hungarian algorithm. Unlike our work, the methods described above use inertial data mostly for identification purposes; they are not used for identification and positioning. In addition, they are specifically designed having only one sensor modality in mind and thus they do not provide a general multi-sensor multi-target tracking framework.

9. CONCLUSION

In this paper we proposed a multi-modal positioning system for highly dynamic environments. We showed that it is possible to adapt Rao-Blackwellised particle filters - traditionally used to discern tracks using anonymous measurements - in order to both identify and track people being monitored by CCTV and holding mobile devices. We further showed that there is significant scope for automatically training the various sensor modalities, and this proved particularly useful in rapidly changing environments. Our experiments showed that even without training, our online approach achieves similar positioning accuracy to the existing off line RAVEL approach; with training the positioning error is decreased by a further 50%.

10. ACKNOWLEDGMENT

We would like to thank Laing O'Rourke for allowing us to conduct our experiments in their construction site and also for funding this research. We also would like to thank the anonymous reviewers and our shepherd for their valuable feedback.

11. REFERENCES

[1] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, Jan 2004.

[2] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House, 1999.

[3] F. Bourgeois and J.-C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM*, 14(12):802–804, Dec. 1971.

[4] T. Bouwmans, F. El Baf, and B. Vachon. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.

[5] A. Doucet, N. De Freitas, and N. Gordon. *Sequential monte carlo methods in practice*. Springer-Verlag, 2001.

[6] A. Doucet, N. d. Freitas, K. P. Murphy, and S. J. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, pages 176–183, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[7] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, Feb 2008.

[8] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.

[9] C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):791–812, Jul 2002.

[10] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.

[11] D. Jung, T. Teixeira, and A. Savvides. Towards cooperative localization of wearable sensors using accelerometers and cameras. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

[12] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[13] D. Lee, I. Hwang, and S. Oh. Optimus:online persistent tracking and identification of many users for smart spaces. *Machine Vision and Applications*, 25(4):901–917, 2014.

[14] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, S. Sen, and V. Handzinski. Microsoft indoor localization competition: Experiences and lessons learned. *SIGMOBILE Mobile Computation and Communication Review (MC2R)*, October 2014.

[15] R. Mandeljc, S. Kovacic, M. Kristan, and J. Pers. Tracking by identification using computer vision and radio. *Sensors*, 13(1):241–273, 2012.

[16] R. Mandeljc, J. Pers, M. Kristan, and S. Kovacic. Fusion of non-visual modalities into the probabilistic occupancy map framework for person localization. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, pages 1–6, Aug 2011.

[17] R. Mautz. *Indoor positioning technologies*. Habilitation thesis, ETH Zurich, 2012.

[18] J. Ning, L. Zhang, D. Zhang, and C. Wu. Robust mean-shift tracking with corrected background-weighted histogram. *Computer Vision, IET*, 6(1):62–69, January 2012.

[19] S. Papaioannou, H. Wen, A. Markham, and N. Trigoni. Fusion of radio and camera sensor data for accurate indoor positioning. In *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, pages 109–117, Oct 2014.

[20] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[21] V. Renaudin, M. Susi, and G. Lachapelle. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012.

[22] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-blackwellized monte carlo data association for multiple target tracking. In *Proceedings of the seventh international conference on information fusion*, volume 1, pages 583–590. I, 2004.

[23] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2 – 15, 2007. Special Issue on the Seventh International Conference on Information Fusion-Part {II} Seventh International Conference on Information Fusion.

[24] S. Seidel and T. Rappaport. 914 mhz path loss prediction models for indoor wireless communications in multifloored buildings. *Antennas and Propagation, IEEE Transactions on*, 40(2):207–217, 1992.

[25] M. Shah, J. Deng, and B. Woodford. Video background modeling: recent approaches, issues and our proposed techniques. *Machine Vision and Applications*, 25(5):1105–1119, 2014.

[26] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122(0):4 – 21, 2014.

[27] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[28] T. Teixeira, D. Jung, G. Dublon, and A. Savvides. Identifying people in camera networks using wearable accelerometers. In *Proceedings of the 2Nd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '09, pages 20:1–20:8, New York, NY, USA, 2009. ACM.

[29] B. Zhang, J. Teng, J. Zhu, X. Li, D. Xuan, and Y. F. Zheng. Ev-loc: Integrating electronic and visual signals for accurate localization. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 25–34, New York, NY, USA, 2012. ACM.