VM-Tracking: Visual-Motion Sensing Integration for Real-time Human Tracking

Qiang Zhai†, Sihao Ding‡, Xinfeng Li†, Fan Yang†, Jin Teng†, Junda Zhu⋆, Dong Xuan†, Yuan F. Zheng‡ and Wei Zhao⋆
†Department of Computer Science and Engineering, The Ohio State University, USA.
‡Department of Electrical and Computer Engineering, The Ohio State University, USA.
⋆The University of Macau, Macau, China.
{zhaiq, lixinf, yanfan, tengj, xuan}@cse.ohio-state.edu, {dings, zheng}@ece.osu.edu, {jdzhu, weizhao}@umac.mo

***Abstract*—Human tracking in video has many practical applications such as visual guided navigation, assisted living, etc. In such applications, it is necessary to accurately track multiple humans across multiple cameras, subject to real-time constraints. Despite recent advances in visual tracking research, the tracking systems purely relying on visual information fail to meet the accuracy and real-time requirements at the same time. In this paper, we present a novel accurate and real-time human tracking system called VM-Tracking. The system aggregates the information of motion (M) sensor on human, and integrates it with visual (V) data based on physical locations. The system has two key features, i.e. location-based VM fusion and appearance-free tracking, which significantly distinguish itself from other existing human tracking systems. We have implemented the VM-Tracking system and conducted comprehensive experiments on challenging scenarios.**

I. Introduction

Human tracking in video provides a direct and context-rich way for localizing humans and analyzing their behavior [20]. It is the enabling technology to a range of applications including smart surveillance, guided navigation, assisted living, etc [14], [15], [4]. For certain applications such as video surveillance, human objects are often being passively tracked; however, there are a plethora of scenarios where humans may actively participate in the tracking process. The cooperative nature of such applications together with the proliferation of mobile devices, enable the possibility of integrating multiple sensor modalities to improve the tracking performance.

Tracking accuracy and real-time performance are two important performance metrics for practical human tracking systems. Tracking accuracy involves two issues, identity and location. A person needs to be continuously tracked throughout videos despite possible distractions from other moving objects and/or changing environments, i.e. maintaining accurate identity association. Meanwhile, a person's physical location also needs to be obtained up to certain desired accuracy. Real-time performance is concerned with the processing latency of a tracking system. For applications such as smart surveillance and assisted living, low processing latency is desirable for prompt responses to unexpected events.

Unfortunately, the state-of-the-art human tracking methods can hardly meet the accuracy and real-time requirements, especially when multiple cameras are employed to cover an extended area and track many objects:

- Tracking accuracy is undermined by environment uncertainty. Many efforts can achieve pretty good performance for a small area over a short period of time. However, in a larger area over a longer time span, the environment changes, e.g., in lighting or human appearances (especially from different cameras), can significantly lower the tracking accuracy.

- Real-time performance is hampered by intensive visual feature computation. Conventional visual tracking algorithms rely on visual features to associate persons across different frames to form continuous trajectories. Appearance features such as histograms, wavelet coefficients, textures, etc. are of high-dimension and involve intensive computation. Moreover, optimally associating humans across two frames is of a cubic computational complexity (in terms of the number of humans), while associating across three or more frames is an NP-hard problem [1].

A. Novelty of Our Approach

In this paper, we propose a novel human tracking system named VM-Tracking, which works by closely integrating visual data (V) with motion sensor information (M). The new system can effectively address the accuracy and real-time issues conventionally associated with visual tracking. There are some existing efforts [21], [22] on integrating visual and motion sensing for human tracking, however, our system is significantly different from them in the following two ways:

- *Location-based VM fusion v.s. Trajectory-based VM fusion.* Existing efforts integrate visual and motion sensing information based on the similarity of V trajectory and M trajectory. They need to record and track long enough trajectories to ensure the distinguishability among different human objects, which leads to long and unpredictable latency.

This work is partially supported by the U.S. National Science Foundation (NSF) grants CNS-1065136, CNS-1218876, the Macau Science and Technology Development Fund under Grant FDCT 023/2013/A1, and University of Macau Research Council under Startup Grant. Any opinions, findings, conclusions, and recommendations in these publications are those of the authors and do not necessarily reflect the views of the funding agencies.

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

Existing work [22] needs 4.5 seconds to correctly associate V and M trajectories. We propose an efficient visual and motion sensing fusion method based on location proximity. Our VM association is performed every 0.3 second. At each time point, our system matches each visual human object with the motion sensor that shows up at the closest location, and updates the V-M locations in a timely manner. Furthermore, our VM fusion is performed individually, which avoids the inefficiency of global matching that compares all visual human objects and motion sensors. However, the accuracy of our location-based VM fusion is more sensitive to the noise of motion sensor due to the accumulative error. To diminish it, we utilize the robust appearance-free V tracking mentioned below. It accurately tracks every human's location at every time point, which is a benchmark to significantly reduce the accumulative error.

- *Human appearance-free V tracking v.s. Human appearance-based V tracking.* Existing efforts rely on the conventional visual tracking, which tracks humans based on appearance similarities. However, it is error-prone to varying camera views and environments. We propose a human shape based visual human tracking algorithm. Instead of each human's specific visual appearance feature like the color ratios, our approach only utilizes the general human shape information which is much less affected by camera views and environments. We call this method appearance-free in this paper. At each time point, i.e. every 0.3 second, our system detects all human objects in camera views based on their shapes, and calculates their physical locations with calibrated cameras. Since humans are more likely to keep similar velocities within a short time period (around 0.3 second in our sytem), our system predicts each human's location based on his previous trajectory and then associates detected human objects with humans based on location proximity. Since we only utilize a general human shape model, our appearance-free tracking algorithm is robust to varying camera views and environments. However, our system needs to search over dense image locations and multiple scales to detect human objects in every 0.3 second, which is potentially a huge computation burden. We avoid this exhaustive search based on the timely location-based VM fusion discussed in the above paragraph. The objects' locations and sizes in the image can be predicted by associated motion sensor, which significantly reduces the searching workload.

B. Our Contributions

With the above two key features, i.e. location-based VM fusion and appearance-free V tracking, our VM-Tracking system can achieve real-time and accurate human tracking. This paper presents a detailed discussion on the design, implementation and evaluation of the VM-Tracking system. We claim the following main contributions of this work:

- We propose an appearance-free visual human tracking system that is the first to efficiently and accurately track multiple humans over a large area covered by multiple cameras.
- We propose an efficient visual-motion sensing integration approach, and design VM-Tracking system with three key modules: location-based human detection, appearance-free object association, and tracking loss recovery.
- We implement the VM-Tracking system with further boost and enhancement, including processing modules pipelining, GPU accelerated human detection and electronic check-point.
- We evaluate system performance with various system settings in terms of accuracy and time, as well as the performance under realistic tracking scenarios. Our tracking system is able to achieve real-time, with less than 0.5 second delay and 0.43 meter error.

II. Related Work

Human tracking has been a hot research area in recent years. Multiple categories of techniques can be applied to human tracking. In this section, we briefly review the works closely related to our VM tracking system.

Vision based tracking can be categorized into two classes: sampling-based tracking such as [18], [10] and tracking-by-detection, like [3], [20]. Shen et. al. [20] proposed a method for camera networks and realized a single object tracking system. Yu et. al. proposed in [26] a system for localizing and tracking multiple people by integrating several visual cues. Efforts are made in applying a-priori constraints to tracking by [10]. Electronics-based technologies localize wireless devices at separate time instants, and then put locations together as tracking results [16]. Banerjee et al. [2] proposed to combine Bluetooth and WiFi RSSI readings to localize cellphones. Yang et al. [25] proposed an indoor localization system based on WiFi fingerprints without heavy human intervention. Sen et al. [19] explored Channel Frequency Response as fingerprints for indoor localization. All these systems suffer from noises from environment and human bodies [28] to some extent. Acoustic localization techniques [13], [9] can achieve high accuracy under the condition that line-of-sight paths exist between sender devices and receiver devices. Meanwhile, three or more anchor devices are required to cover the same area to perform time-of-arrival (TOA)-based trilateration.

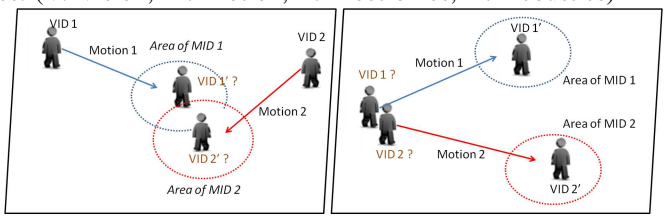
Methods utilizing data fusion of different types of sensors are also found in literature. Teixeira et. al. [21], [22] achieve tracking by theoretically solving the problem of trajectory association of vision and motion-sensor, however, assuming that vision algorithm can perfectly detect all human in the view and motion sensors are noiseless. EV-Loc [27] integrates electronic and visual signals for localization. It requires one-to-one matching between electronic devices and visual detections and accumulating E-V data to find the correct associations, which is less error-tolerant and not a real-time approach as ours. EV-Human [12] is the extension of EV-Loc and has the same problems. Fan et. al. [11] proposed a particle-filtering based motion sensor fusion approach for self-tracking. A combination of inertial sensor and camera for self-tracking is introduced in [8]. Roetenberg et. al. [17] proposed a system that consists of magnetic sensors and inertial sensors for

Methodology	Insensitive to Human Body Interference	Insensitive to Environment	Commodity Hardware	Median Accuracy	Latency
Location based V-M fusion	✓	✓	✓	High (0.43m)	Low, Real-time
Trajectory based V-M fusion [21], [22]	✓	×	✓	Medium (0.8m)	High
Doppler effect based M-A fusion [9]	×	×	×	High (0.4m) ∼ Medium (0.92m)	Low, Real-time
Appearance based V [26], [18]	✓	×	✓	Medium (0.963m)	High
E Antenna array [24]	✓	✓	×	High (0.23m)	Low, Real-time
E RSSI Fingerprinting [25]	×	×	✓	Low (Room-level)	Medium

TABLE I: Comparison of our system with related techniques. (V: Vision, M: Motion, E: Electronics, A: Acoustics)

motion tracking. NavShoe system [7] is an orientation only tracking system designed to embed in shoes based on wireless inertial sensors and achieves meter-level accuracy.

We summarize the differences of our work with existing works in Table I. We can see that our system strikes a good balance among performance and scenario compatibility. With a moderate cost of infrastructure, we are able to achieve high accuracy and real-time performance for dense human tracking.



(a) Objects come close (b) Objects go away

Fig. 1: V-M integration for human tracking

III. VM-Tracking System Design

In this section, we present the design of our VM-Tracking system. We start with the rationale of VM-Tracking, and then introduce the workflow of our system, followed by the detail of primary system modules.

A. Design Rationale

The objective of our system is to accurately track multiple human objects in real-time within a large space. Despite the recent advance in visual tracking research, the accuracy and real-time performance of the state-of-the-art visual tracking algorithm is still far from perfect [23]. Visual tracking of multiple human objects faces even greater challenges when we have multiple cameras for a large area, as required in our targeted application. For our application, we argue that camera-only tracking techniques cannot meet the accuracy and real-time requirement at the same time, based on the following observations: 1) Visual appearances of the same object across different camera views may significantly differ, which requires complicated and hence heavy computations to handle properly; 2) Visual occlusions can hardly be resolved with a single camera, which may cause erroneous tracking. Employing multiple cameras with overlapped field-of-view to resolve occlusions introduces large computation burden, and thus hampers real-time performance. 3) Appearance models are usually high-dimensional, resulting in high computation complexity. Existing VM fusion approaches [21], [22] that rely on conventional visual tracking also suffer from the same problem. Besides, their trajectory-based VM fusion is inefficient because recording and tracking unpredictably long trajectories are required to ensure the distinguishability among different human objects.

In this paper, we incorporate the motion sensor measurements of a mobile device as a new measurement dimension, in order to address the dilemma of accuracy and real-time inherent to the existing tracking methods. Visual and motion sensor information is closely integrated in our novel tracking methodology named VM-Tracking. Fusing all motion sensor readings enables us to estimate the user's instantaneous velocity and location. The integration of visual and motion sensor information as proposed in this paper overcomes the accuracy and real-time problem faced by conventional visual tracking and existing VM fusion tracking systems due to the following three reasons.

First, human detection from individual video frames can be greatly accelerated with the aid of motion sensor information. For conventional visual and existing VM fusion approaches, detecting human objects from a video frame requires searching over dense image locations and multiple scales, and therefore, the computation is intensive. With the motion sensor information, exhaustive search is not necessary. Given the previous human detections in a video frame and their motion estimated from the motion sensor readings, the objects' locations as well as their image sizes in the frame can be predicted, as the dashed circles shown in Figure 1. Only local search around the predicted image regions and limited image scales is required for finding the human objects in the current frame. The reduced searching space improves real-time performance without loss of accuracy.

Secondly, location-based visual and motion sensor integration provides an effective solution to track human objects across different frames. Human objects are considered as multiple moving points based on their physical locations. We choose to use motion sensor information as the evidence for associating multiple human detections that belong to the same

person across different frames. In light of the spatial limitations of human objects, our tracking system utilizes physical location proximity and velocity to measure the likelihood for associating human detections. Figure 1 illustrates two such cases. The physical location based association enables our system to track each person individually, which is more scalable than trajectory-based VM fusion algorithms.

Thirdly, tracking loss can be recovered with motion sensor information. In practical scenarios, it is common for tracked objects to undergo long visual occlusions. With motion sensor information, the locations of human objects can be continuously acquired. Similar locations of a visual object can be associated to form a continuous tracking result.

B. Workflow

Figure 2 shows the workflow of our VM-Tracking system. Different tracking procedures are illustrated using different colors and line types.

First, we have multiple video cameras covering a large area with very small overlapped areas to form a continuous tracking area. Our motion data come from the motion sensors of users' mobile phones and are denoised locally. All the V and M data are collected by a central server for further processing. We identify the motion data coming from the same mobile phone with an electronic identifier such as WiFi MAC address, called MID. Meanwhile, we have detected visual objects in the video frames, called VID. Our VM-integrated human tracking is based on the association of the MIDs and VIDs. The association uses physical proximity.

The MID location is estimated by its location in the last frame and its motion during the current frame. If no previous location is available, it means this MID has not been initialized with an VID or it has got lost. We will discuss the initialization/loss recovery module later (in green dashed lines). On the V side, human detection is performed on the current video frame with the acceleration of motion information. If there are no detected VIDs (for example caused by occlusion), we simply use the estimated MID locations as the tracking results in the current frame, and then go to the next frame (in orange dotted lines). If there are detected VIDs, we compute their physical locations by calibrated cameras. Next, we integrate the MID and VID based on the physical proximity. Once the similarity of a VID and a MID exceeds a certain threshold, we consider them a valid association and update their locations using the VID location. After the update, we continue to the next frame.

Besides the normal tracking module, we have another workflow for the initialization/loss recovery module (denoted as the green paths in Figure 2). We determine a tracking loss under the condition of no valid VID-MID association. The initialization is same to the loss recovery, whose purpose is to find a correct VID for a "new" MID. Our initialization/loss recovery is based on V-M association filtering. After assuming a large candidate VID set, we gradually filter away impossible associations and finally reach the unique correct VID with theoretical guarantee.

Next, we discuss detailed designs of three key modules: human detection, object association and loss recovery.

C. Motion Sensor Accelerated Human Detection

In the proposed VM-Tracking system, the instantaneous locations of tracked objects are obtained using visual human detection accelerated by motion sensor information. The human objects are delineated by bounding boxes in a video frame, and their physical locations can be easily calculated using projective geometry model given calibrated cameras. Note that such visual detections are yet to be associated with individual object identities in order to form continuous trajectory, which will be described in detail in the following subsection.

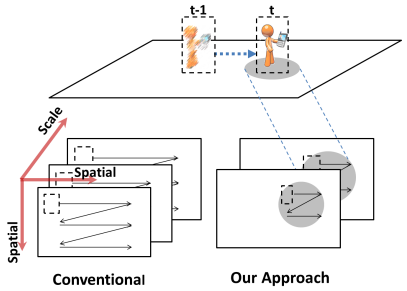


Fig. 3: Motion sensors accelerated human detection

Detection performance of state-of-the-art visual human detection algorithms has now reached some level of maturity, in that human objects can be reliably detected from nature scenes with high detection and low false alarm rates [5], [6]. However, due to the nature of the underlying computation model, visual detection often incurs significant computational cost. The visual detection process involves exhaustive search, i.e. sliding a window across an image at multiple scales and classifying each local window as the target or background. For practical applications, video data from multiple cameras need to be processed simultaneously in real-time, which imposes heavy computational demand on the video system.

We propose to utilize motion sensor information to avoid exhaustive search, and thus provide a computationally efficient solution to visual human detection under cooperative settings. Motion sensor information from a target is used to predict its motion between two consecutive frames. Given a target's previous location, the possible area within where it is likely to appear can be determined based on the error model of motion sensors. The corresponding region and size in the image domain are determined given the camera calibration model. Sliding window search can now be confined within a local neighborhood of the spatial-scale space, as illustrated in Figure 3.

The image pyramids are images at a sequence of different scales for multi-scale object detection. Usually the detection goes through all the pyramid levels. However, the largest scale and the smallest scale cost very different times. So it becomes necessary to save time on large scales if they do not probably contain the detection results. With the motion information,

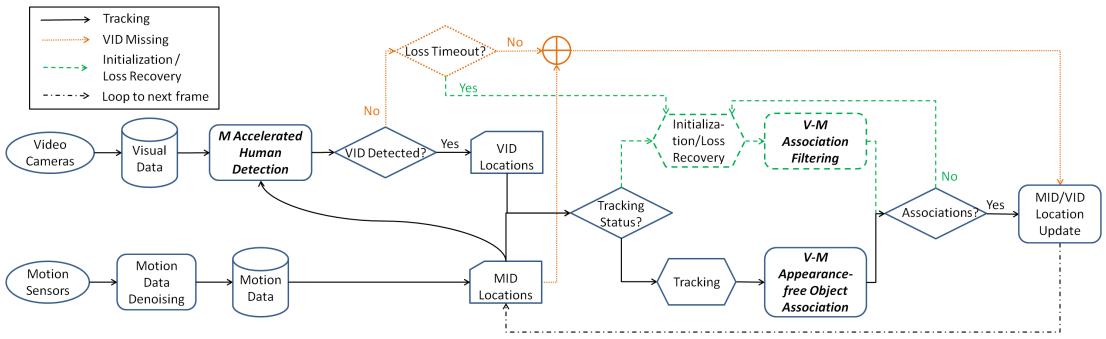


Fig. 2: VM-Tracking system workflow

we reduce the number of pyramid levels for detection by predicting the possible pyramid levels the target appears in detectable scales. To estimate the pyramid level at which the detection performs, we asume H is the homogenous transform matrix obtained from camera calibration. At time $t-1$, the left bottom corner and right bottom corner of the bounding box is denoted as x_0 and x_1. The width of the bounding box is w. Then the phisical locations of them in real world is computed as $X_0 = x_0 \cdot H$ and $X_1 = x_1 \cdot H$, respectively. Their next location X_0' and X_1' at time t is predicted using the motion information provided by motion sensor. The predicted real world locations are mapped back to image plane in pixel coordinates, from $x_0' = X_0' \cdot H^{-1}$ and $x_1' = X_1' \cdot H^{-1}$. Then the width w' of the new predicted bounding box is estimated as the Euclidean distance between x_0' and x_1'. Based on the error model of motion sensors, this value may vary in $[w'-n, w'+n]$, in which n is the noise introduced by the error. The ratio of w and $w' \pm n$ is the estimated range of ratio of current pyramid scale and predicted pyramid scale.

D. Motion Sensor Assisted Appearance-free Object Association

We utilize the location proximity to associate human detections at consecutive time steps based on motion sensor information. For consecutive frames $t-1$ and t, multiple persons are visually detected and their physical locations can be calculated given the camera's calibration data. We represent the set of visually detected persons at time $t-1$ as $A = \{a_i\}_{i=1,\ldots,M}$, and at t as $B = \{b_j\}_{j=1,\ldots,N}$, respectively. Further denote the physical location of the i-th person seen from the visual camera at time $t-1$ as $v(a_i)$, and similarly the j-th at time t as $v(b_j)$. For a_i at time $t-1$, we can know his speed from the visual cameras, as well as from the motion sensor. On the visual camera side, we can estimate his moving speed, and calculate his estimated location at time t, which is denoted as $\tilde{v}_t(a_i)$. Similary, on the v sensor side, we can use acceleration information to estimate his average moving speed and calculate his estimated location at t, denoted as $\tilde{m}_t(a_i)$. Assume we know the actual speed distribution, then we can calculate q_{ij} (the matching probability of a_i and b_j based on visual estimations) and r_{ij} (the matching probability of a_i and b_j based on motion estimations) as

$$q_{ij} = (a_i = b_j | \tilde{v}_t(a_i), v_t(b_j))$$
$$r_{ij} = (a_i = b_j | \tilde{m}_t(a_i), v_t(b_j))$$

Suppose the visual estimation has a standard deviation of σ_v, and the motion estimation has a standard deviation of σ_m, we calculate the fused probability p_{ij} as

$$p_{ij} = q_{ij}^{\sigma_v^2/(\sigma_v^2+\sigma_m^2)} \cdot r_{ij}^{\sigma_m^2/(\sigma_v^2+\sigma_m^2)} \quad (1)$$

With the above probabilty, we can gradually filter away impossible matching based on a threshold learnt offline. We perform the filtering over rounds until a one-to-one matching result is found.

When multiple objects are in close proximity or mutually occluded, we resort to the motion sensor information for resolving the ambiguity arising from visual domain. When two persons approach each other and undergo visual occlusion within a camera's view, the visual detection algorithm will fail to generate two separated detections. Conventional visual tracking methods will very likely fail, and may produce switched target identities after they separate, if two objects are visually similar. With the aid of motion sensor, our method can handle the occlusion case. Once the objects separate and produce multiple visual detections, the motion direction as estimated from motion sensors serves as the hint for associating a visual detection to the one in the previous time step, as illustrated in Figure 4.

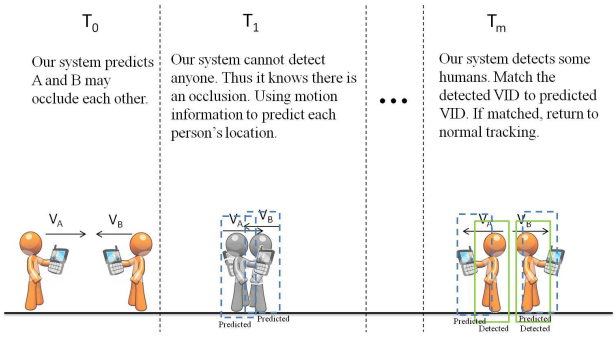


Fig. 4: Motion sensors solve occlusion problem

E. Motion Sensor Guided Loss Recovery

Ideally, our system keeps tracking the human objects all the time. However, tracking losses could happen due to many complex reasons. It is not easy for existing visual tracking methods to handle this issue. Trajectory-based VM tracking algorithms have to start over, which significantly hurts their

real-time performance. In this light, we propose a location-based loss recovery mechanism. When a MID loses its associated VID, our tracking system starts the tracking loss recovery module by assigning this MID to all possible VIDs in the view. The idea is to filter out over time impossible associations with continous V-M tracking until the only correct VID is left.

On a more technical level, we have several sequences of visual locations over time (from different persons) from the cameras, and a sequence of acceleration readings (on one person) from the motion sensors. We want to tell if one of those visual sequences corresponds to the motion sequence. For one person, we get a sequence of velocities from the visual side. Denote it as $\{P_i\}$, and P_i is calculated between time point T_{i-1} and T_i. On the other hand, we integrate accelerations on the M side to get the velocity sequence for a specific MID, and denote it as $\{Q_i\}$. Q_i is integrated from T_{i-1} to T_i. It is worth noting that $\{P_i\}$ and $\{Q_i\}$ are noisy data. The noise of $\{P_i\}$ and $\{Q_i\}$ can be considered i.i.d. Normally, these noise can be modeled as Gaussian. If $\{P_i\}$ and $\{Q_i\}$ are generated by different persons, their average difference will be strictly greater than zero after enough observations. In this situation, we have $E[P_i - Q_i] > 0$. We denote $\overline{P_i - Q_i} = \frac{\sum_{i=1}^{n} P_i - Q_i}{n}$. We have the following theorem.

Theorem 1: For two sequences $\{P_i\}$ and $\{Q_i\}$ with P_is and Q_is bounded, $E[P_i] > E[Q_i]$, $\forall c \in (0,1), \forall \varepsilon \in (0, E[P_i] - E[Q_i])$: $\exists N$,

$$P\left(\overline{P_i - Q_i} > \varepsilon\right) > c, (n \geq N). \tag{2}$$

Theorem states that if the V and M observations are from different people, we will eventually be able to tell with enough observations. If we model the noises of V and M observations as Gaussian, we will be able to accurately tell how much confidence we have of telling them apart (ε and c) after how many rounds (N). Due to space limitation, we skip the proof of Theorem 1 and detailed calculations.

Theorem 1 gives the theoretical foundation that we are able to associate lost VIDs with correct MIDs. However, there is still another problem, the state explosion problem. We will match MIDs with every VID in every frame. In an ideal scenario where nobody enters or leaves, suppose we have D frames, and N MIDs. We will have a total number of combination at N^D. This will significantly hurt our real-time performance as D gets larger. Our location-based object association method actually solves this problem efficiently. Motion sensors also give the displacement constraint of each human object over a fixed time period. Assume the association threshold is τ. We can guarantee that no VIDs that exceed τ will be filtered away in the course, because sub-trajectories carry larger probabilities than the entire trajectory. The following lemma and theorem are self-evident.

Lemma 1: $\prod_{i=1}^{n} P_i \leq \prod_{i=a}^{b} P_i$ $(1 \leq a \leq b \leq n)$.
where P_is are all probabilities, i.e., $0 \leq P_i \leq 1$. Theorem 2 follows.

Theorem 2: If a complete trajectory's probability exceeds τ, any sub-trajectory of this trajectory will have a probability exceeding τ.

IV. Implementation and Evaluation

In this section, we present our system implementation, report our experiment results on VM-Tracking system and show the system's performance on accuracy and time.

A. Implementation

The implementation of our system has two main components: a front end that collects and transmits video images (V) and motion sensor data (M); a back end that receives V and M data and performs human tracking.

We use commodity cameras (D-LINK DCS-930L) to shoot the area of interest. We set the V data rate to 3 frames per second which is a very low video rate to demonstrate the efficiency and robustness of our system. We implement an Android application to collect and transmit motion sensor data from users' mobile phones (Nexus S). The M data rate is set to 5 readings per second. Raw motion sensor data contains huge noise that causes error accumulation and tracking derailment. In our system, we apply Google's denoising technique (Rotation Vector Sensor) to improve the accuracy of M data based on the rotation vector sensor on mobile phones.

Our back end server is equipped with an Intel Xeon E5 CPU, two NVIDIA GTX 760 GPUs and 8 GB memory. The received M data are inserted into a MySQL database. For the V data, we perform human detection on the frames and compute physical locations of the detected VIDs with a state-of-art algorithm called DPM [6]. Then, VM-tracking is executed based on the V and M data in database, following the workflow as we discussed before. Finally, we implement a Java GUI to show video frames live with the human objects marked with associated EIDs.

Next, we highlight three key components we implemented for system efficiency:

- **Processing Modules Pipelining.** We pipeline the whole tracking process where every module greedily starts working once its inputs are ready. The real-time performance depends on the slowest module (which is usually the human detection). If the slowest module is faster than the V and M data input speed (1/3 seconds), we say our system is real-time. In addition, the tracking delay of our system is determined by the critical path of the tracking process, which is along the V data processing modules actually.

- **GPU Accelerated Human Detection.** We utilize GPU to accelerate human detection. Specifically, we accelerate three steps of the human detection algorithm (i.e., DPM in our system): pyramid building, feature generation and filtering. These three steps take 99% of the overall detection time. We notice that these steps are parallelizable, so we can take advantage of GPUs on parallel computing.

- **Electronic Check-point.** We employ electronic check-points to associate an MID with a VID to initialzie tracking based on the observation that when a person approaches a wireless sensor (e.g. AP), say less than 1 meter away, the RSSI readings of his mobile phone are strong and robust against

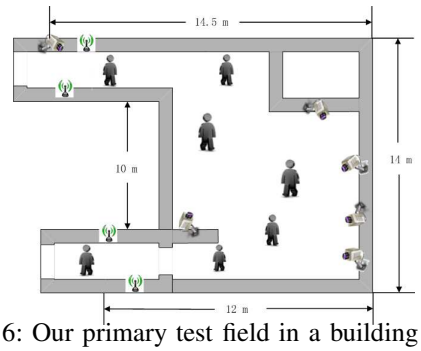


Fig. 6: Our primary test field in a building lobby

environmental noise. So we directly associate that phone with the person standing near the AP that is observed by our cameras. The small area around the AP becomes a check-point.

B. Experiment Setup

We have set up two different test fields to evaluate our system: a large lobby (Figure 6) and a small office. The large one has a complex background and non-uniform illumination. Six cameras were deployed to cover the whole area. The small one has static lights with two cameras deployed. We set up electronic check-points at the entries of these two test fields. We had 5 participants in our experiments for tracking. Each participant took an Android phone in the hand or in the pocket and walked around freely in the experimental area. We also had several other persons to emulate non-users, who were walking around without mobile phones.

We evaluated our system using two metrics: tracking error and processing time. The tracking error is defined as the average distance between a user's trajectory determined by our system and his ground truth trajectory. Due to human detection box drift, there is a small error of the visual localization, which is around 0.35 meter according to our measurement on data samples. The processing time has two aspects: processing delay and update interval. The processing delay is the time that our system spends to update a user's location after the user moves. The update interval refers to the time between two consecutive location updates.

We compared three tracking algorithms: our system (VM), appearance-free tracking (V-AF) and Incremental Visual Tracking (V-IVT) [18]. The VM algorithm is our visual-motion sensing integrated tracking algorithm. The V-AF algorithm is the physical-location-based, appearance-free visual tracking. Its difference from the VM algorithm is that the V-AF algorithm does not use the motion sensor data. The IVT is one representative conventional tracking algorithm in the computer vision.

C. System Performance under Controlled Settings

1) Accuracy: We measure the following factors that could affect our tracking accuracy: user distance, user speed, user number, user appearance and environmental illumination.

– User distance. User distance affects the occlusion degree of the users. We let two users move along two parallel lines in opposite directions back and forth under one camera's view. As the distance between the two lines increases, there is less occlusion. The tracking errors are shown in Table II. When two users are very close to each other (0.6 meter distance) with severe occlusion, our tracking system has a tracking error at around 1 meter. This demonstrates the effectiveness of the motion sensor data on helping distinguish two users with their different moving directions.

User distance	0.6 m	1.8 m	2.4 m
Tracking error	1.08 m	0.38 m	0.35 m

TABLE II: Tracking error under different user distances

User speed (m/s)	0.5	0.7 (slow)	1.0	1.5 (fast)
Tracking error	0.35 m	0.35 m	0.35 m	0.57 m

TABLE III: Tracking error under different user speeds

– User speed. User speed affects the movement of the users in two consecutive frames. We let three users move freely in the large test field with different speeds. The average tracking errors are shown in Table III. We can see that our tracking system never losses tracking of the users with slow or normal speeds. The tracking error slightly increases as the user speed increases. This shows that our appearance-free tracking based on physical location proximity works well with normal human walking speeds.

– User number. User number affects the density of the users in the tracking area. We let different numbers of users move freely in both the small and the large test fields. The results are shown in Table IV. Our system performs well with a dense human crowd, e.g., 3 users in the small field and 5 users in the large field.

– User appearance and illumination. We performed the experiments in both the small and the large test fields (with different illumination). We let the users change their clothes to make their appearances very similar or very different. The results are shown in Table V. Our system never losses track of the users in any of the four test situations. Note that the 0.35 meter error comes from the visual localization noise. In other words, our system is appearance-free and robust to the change of environmental illumination.

2) Time: We measure the following factors that could affect the real-time performance: user number, and scale of camera's view.

– User number. User number affects the processing time of human detection, which is slowest module in our VM tracking

User number (small field)	1	2	3
Tracking error	0.35 m	0.35 m	0.40 m
User number (large field)	1	3	5
Tracking error	0.35 m	0.35 m	0.44 m

TABLE IV: Tracking error with different user numbers

	Same Appearances	Different Appearances
Uniform Illumination	0.35 m	0.35 m
Non-uniform Illumination	0.35 m	0.35 m

TABLE V: Tracking error under different user appearances and illumination

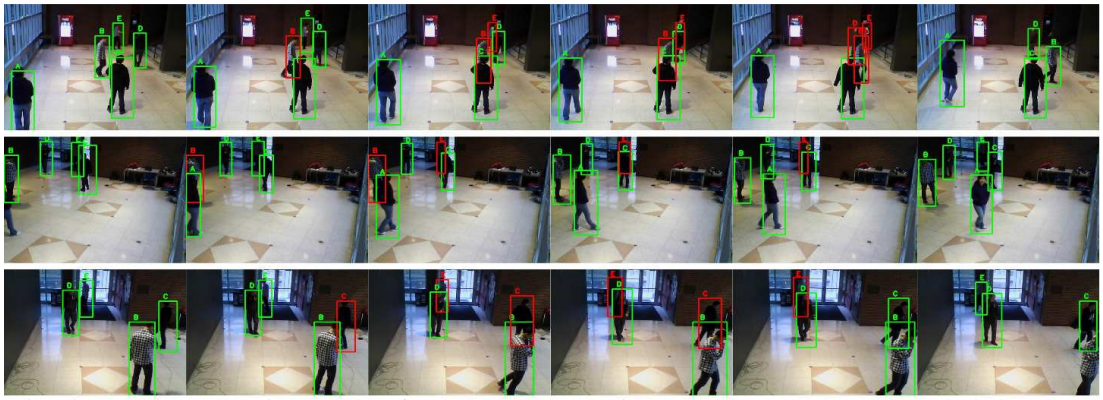


Fig. 5: Tracking scenario snapshot from three camera views (red boxes represent occluded persons)

User number	1	2	3
Processing delay	374 ms	437 ms	512 ms
Human detection time	159 ms	232 ms	273 ms

TABLE VI: Processing time under different user numbers

Camera view	Narrow	Wide
Processing delay	371 ms	480 ms
Human detection time	157 ms	276 ms

TABLE VII: Processing time under different camera views

procedure. Table VI shows the overall processing delay and the human detection time under different user numbers. As user number increases, both processing delay and human detection time increase. But their values are small. As we discussed before, if the slowest module is faster than the data rate (3 fps), we know that our tracking system is real-time.

– Coverage of cameras. The depth of the area covered by the camera directly determines the variation range of a person's size shown in the image, which will have impact on the range of pyramid level selection in motion sensor accelerated human detection. The time performance under different user numbers is shown in Table VII. The smaller the area is covered, the less the human scale changes in the image, thus the time cost will be reduced.

D. System Performance under Realistic Settings

1) Single User Tracking: For the single user tracking experiment, we let one user walk around in the large test field for 5 minutes. We purposely let another two non-users to walk through the test field. We repeated this experiment 5 times. The tracking errors are shown in Figure 7(a).

Our system is able to continuously track the user. There is only one tracking loss in the trace shown in Figure 7(a), which happened at around 245th second. It means we achieve almost 100% correct tracking during the 5 minutes. Furthermore, the tracking loss got recovered soon within 5 seconds. The overall average tracking errors are only 0.42 meter. On the other side, we observe more losses in the V-AF tracking. The recovery of the V-AF tracking usually costs more time. The V-AF tracking may totally lose track, for example, at the 240th second. On the opposite side, our VM tracking can always recover from tracking losses based on the diversity of movement of different

	1-user tracking	5-user tracking
VM	0.42 m	0.43 m
V-AF	2.33 m	3.50 m
V-IVT	4.44 m	3.96 m

TABLE VIII: Tracking errors of different tracking algorithms

	1-user tracking	5-user tracking
VM	475 ms	472 ms
V-AF	579 ms	654 ms
V-IVT	870 ms	928 ms

TABLE IX: Processing delay of different tracking algorithms

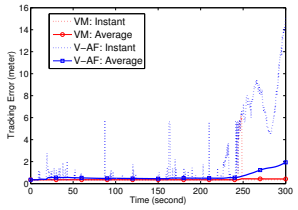
persons. The V-IVT performs the worst. It usually got lost in about 20 $\sim$ 30 frames (about 10 seconds), and can not get recovered from the tracking losses.

For the time performance, we summarize the processing delay in Table IX. The improvement by utilizing motion sensor data is about 100 ms. Our VM tracking is real-time. The detailed processing times of individual tracking components are similar to the multiple user tracking case (as shown in Figure 8). We will discuss it in detail later.

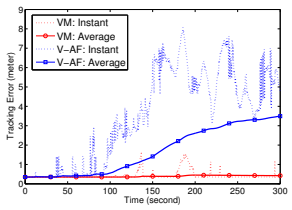
2) Multiple User Tracking: For the multiple user tracking experiment, we let 5 users walk around in the large test field for 5 minutes. The area was dense with 5 users, and some occlusions are shown in Figure 5. We repeated this experiment 5 times. The tracking errors are shown in Figure 7(b).

We can see that more tracking losses happened in the 5-user tracking case. However, our VM tracking still performs very well in the 5-user tracking. The overall average tracking error is similar to that in the single user tracking case. On the other side, the performance of the V-AF tracking degrades significantly as the number of tracked users increases from 1 to 5. We conclude that our motion-assisted tracking system is much more robust with a dense human crowd. We also show the cumulated density function of the successful tracked time (i.e. the first loss time) of V-IVT, V-AF and our VM tracking in Figure 9. The VM algorithm has a much longer successful tracking duration. Also note that even after tracking lose takes place, VM has ability to recover.

The time performance of our VM tracking is shown in Figure 8. First, the processing delay is relatively stable during



(a) Single user tracking



(b) Five user tracking

Fig. 7: Tracking errors of VM and V-AF tracking

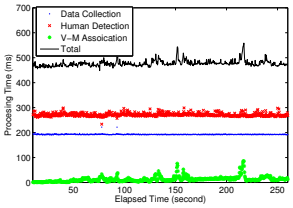


Fig. 8: Processing time of VM tracking (five user tracking)

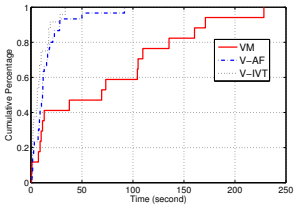


Fig. 9: CDF of successful tracking time

the 5 minutes period, with the average value at 472 ms. Second, the slowest component (human detection) keeps below 300 ms, which is smaller than the frame interval of 333 ms. With our pipelined processing mechanism, our VM tracking is real-time. On the other side, we observe the V-AF tracking suffers from the dense crowd and V-IVT has the highest processing delay.

V. Conclusion

In this paper, we presented an accurate and real-time human tracking system by integrating visual camera and motion sensor data. We proposed an appearance-free tracking algorithm and a physical location based VM fusion algorithm to track visual human objects with the assistance of their motion sensors. They significantly mitigate the dependency of long trajectories, high computation overhead of video processing and the occlusion problem, and thereby provide continuous and accurate tracking results. We have conducted comprehensive experiments based on large scale real-world implementation. The results show the superior performance of our system in terms of time efficiency and tracking accuracy.

References

[1] C. Arora and A. Globerson. Higher order matching for consistent multiple target tracking. In *ICCV*, 2013.

[2] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. D. Corner. Virtual compass: Relative positioning to sense mobile social interactions. In *Pervasive '10*, pages 1–21, 2010.

[3] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, 2011.

[4] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan. Mobile phone-based pervasive fall detection. *Personal and Ubiquitous Computing*, 14(7):633–643, 2010.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 1, pages 886–893. Ieee, 2005.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[7] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25(6):38–46, 2005.

[8] E. Foxlin, L. Naimark, et al. Vis-tracker: A wearable vision-inertial self-tracker. *VR*, 3:199, 2003.

[9] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *INFOCOM, 2014 Proceedings IEEE*, 2014.

[10] Y. Lao, J. Zhu, and Y. F. Zheng. Sequential particle generation for visual tracking. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(9):1365–1378, 2009.

[11] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.

[12] X. Li, J. Teng, Q. Zhai, J. Zhu, D. Xuan, Y. F. Zheng, and W. Zhao. Ev-human: Human localization via visual estimation of body electronic interference. In *Proc. of IEEE INFOCOM Mini*, 2013.

[13] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, 2013.

[14] MarketsandMarkets. Video Surveillance Systems & Services Market - Analysis & Forecast (2013 - 2018). http://goo.gl/MdmIvX, 2013.

[15] V. P. Munishwar, V. Kolar, and N. B. Abu-Ghazaleh. Coverage in visual sensor networks with pan-tilt-zoom cameras: the maxfov problem. In *Proc. of IEEE INFOCOM*, 2014.

[16] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang. A study on object tracking quality under probabilistic coverage in sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(1):73–76, 2005.

[17] D. Roetenberg, P. J. Slycke, and P. H. Veltink. Ambulatory position and orientation tracking fusing magnetic and inertial sensing. *Biomedical Engineering, IEEE Transactions on*, 54(5):883–890, 2007.

[18] D. Ross, J. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Computer Vision-ECCV 2004*, pages 470–482. Springer, 2004.

[19] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: spot localization using phy layer information. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 183–196. ACM, 2012.

[20] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C. T. Chou. Efficient background subtraction for real-time tracking in embedded camera networks. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 295–308. ACM, 2012.

[21] T. Teixeira, D. Jung, and A. Savvides. Pem-id: Identifying people by gait-matching using cameras and wearable accelerometers. In *Proceedings of International conference on Distributed Smart Cameras*. ACM, 2009.

[22] T. Teixeira, D. Jung, and A. Savvides. Tasking networked cctv cameras and mobile phones to identify and localize multiple people. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 213–222. ACM, 2010.

[23] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proc. of CVPR*, pages 2411–2418. IEEE, 2013.

[24] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *NSDI, 10th USENIX Symposium on Networked Systems Design and Implementation*, 2013.

[25] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 269–280. ACM, 2012.

[26] S.-I. Yu, Y. Yang, and A. Hauptmann. Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3714–3720. IEEE, 2013.

[27] B. Zhang, J. Teng, J. Zhu, X. Li, D. Xuan, and Y. F. Zheng. EVLoc: Integrating electronic and visual signals for accurate localization. In *Proc. of ACM MobiHoc*, 2012.

[28] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng. I am the antenna: accurate outdoor AP location using smartphones. In *Proceedings of ACM MobiCom*, pages 109–120, 2011.