Vision-Track: Vision based Indoor Tracking in Anchor-free Regions

Gopi Krishna Tummala, Rupam Kundu, Prasun Sinha, Rajiv Ramnath
Department of Computer Science and Engineering
The Ohio State University, Columbus, Ohio - 43202
tummala.10@osu.edu, kundu.24@osu.edu, sinha.43@osu.edu, ramnath.6@osu.edu

ABSTRACT

Smart-devices can render high quality location services when endowed with the ability to analyze information conveyed through video feed. In this paper, we aim to provide tracking services by using a mobile smart camera such as in google glasses and smartphones considering the following three objectives: (1) *No additional deployment*, (2) *No user-side instrumentation or hardware upgrades*, and (3) *Easy adoption in practice*. Existing RF or VLC based solutions for indoor tracking can provide location and orientation only when there are dense deployments of APs or VLC bulbs (anchor points) in user's field of view. Vision-Track is the first vision based solution that can track the camera's location and orientation indoors even when no anchor point is in line-of-sight (LOS). Vision-Track deployed in an indoor college building provides a median localization accuracy of 49 cm.

CCS Concepts

•**Hardware** → **Wireless devices;** *Sensor applications and deployments;* •**Computer systems organization** → *Real-time systems;*

Keywords

Indoor Tracking, Computer Vision, Experimentation.

1. INTRODUCTION

Camera-based assistance for day-to-day human activity is opening up a new horizon with the advent of smart-glasses, enhanced smartphone cameras, intelligent security cameras, etc. Popular commercialized smart-glasses like Google glasses and Microsoft's HoloLens can record daily activities and further analyze them to make intelligent decisions. These features when integrated with the knowledge of location, can benefit many applications associated with indoor navigation such as tracking, vision aid for visually impaired people, object tagging, security and activity monitoring. Crowd-sourced video associated with accurate location coordinates can be used to create holistic views of buildings and events can be used by virtual reality based applications. In addition, the location services can enhance customized smart-devices for performing stereo-vision, object scanning and thermal imaging.

Both location and orientation are important for different applications like gaming, virtual reality, indoor navigation, etc. where precise directionality is needed. Conventional WiFi based localization techniques [13, 3, 14, 9] can provide indoor localization with an error of 2-3 m but fails to provide the orientation of the user. A small subset of the RF-localization solutions like Ubicarse [4] can provide both location and orientation. But, it requires the APs to be in LOS of the user. Therefore, to provide continuous location services in indoor settings, RF-localization requires dense deployments of APs in the user's field of view, which entails expensive modifications to existing infrastructure. Fingerprinting based approaches [11] require offline training and have a meter level error margin. Inertial sensors can measure location and orientation but they are known to suffer from error accumulation [15]. Recent works on VLC (Visible Light Communication) [2, 6, 5] encode data in terms of light intensity changes imperceptible to human eyes but can be decoded successfully using light sensors. VLC can enable users with accurate localization services akin to GPS in outdoor environments, only if the VLC-bulbs are closely spaced. So the key question is - *How can we achieve accurate location and orientation information when no anchor point (AP or VLC-bulb) is in LOS?*

Vision-Track is the first vision based solution that can continue to track the camera's location and orientation even when no anchor point is in sight. Thus it is specifically designed with sparse deployment in mind. The system is implemented as a smart-glass app that makes use of the camera to record video and perform real-time computation. Vision-Track user starts with initial location and orientation obtained through standard VLC or RF based localization techniques. A smart-device with a single camera is either carried in person or attached to a moving cart (like a battery operated passenger carrying cart inside an airport). The camera captures a video of the scene ahead in a way such that the ceiling is included in the camera's view. The user will be tracked in real-time for stretches where no anchor point is visible.

The system is tested extensively in college buildings. Experimental evaluations demonstrate a 49 cm median error for Vision-Track. The contributions are as follows: (i) The *Vision-Track* solution that can accurately track the camera's

HotWireless'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2980115.2980135

movement and orientation starting from a known location and orientation. (ii) Extensive experimental evaluation.

2. CHALLENGES

To develop a practically usable solution, the following challenges need to be addressed:

Lack of depth information: The video feed captured by commodity smart devices lacks depth information of the captured objects. In addition, the association of the observed objects with their real entities is not evident from the video. **Arbitrary camera movement:** The camera may be subject to arbitrary motions and wobbling due to human factor or cart's arbitrary movements. **Identifying points in the ceiling:** The points that are tracked by Vision-Track to derive the relative motion of the camera must be at a known height. The ceiling points are desirable for this purpose. But Vision-Track needs to identify among a set of points which ones correspond to the ceiling which is a non-trivial problem. **Error accumulation in inertial sensors:** Error accumulation in smartphone's sensors, like gyroscope or accelerometer, inflates with time.

3. THE CAMERA MODEL

The *Pinhole camera model* [12] describes the geometric relationship between the 2D image-plane (i.e, pixel positions in a camera capture) and the 3D ground coordinate system. Let the image plane be represented by the UV-plane and the camera coordinate system be represented by the (XYZ) space. Let us assume that the perpendicular ray emanating from the center of the camera frame is along the Z-axis, V-axis is parallel to Y-axis and U-axis is parallel to X-axis (see Figure 1). The origin of the (XYZ) space is shifted from the image plane by a distance equal to the focal length f of the camera. Let the point $(x_1,\ y_1,\ z_1)$ correspond to $(u_1,\ v_1)$ pixel location in the image plane. The geometrical relationship between the two coordinate systems, using the pinhole model is given by,

$$\begin{aligned} \frac{u_1}{f_u} &= \frac{x_1}{z_1}, \\ \frac{v_1}{f_v} &= \frac{y_1}{z_1}, \end{aligned} \tag{1}$$

where f_u and f_v are the focal lengths of the camera[1]. The

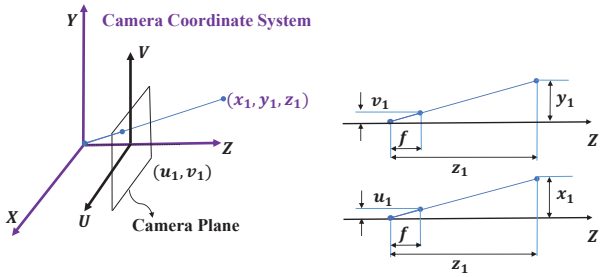

Figure 1: ***Geometric relationship between image plane and the camera coordinate system. Illustration assumes*** $f = f_u = f_v$

[1]The ideal depiction in Figure 1 shows them to be same, but the model allows them to have different values

pinhole model translates each pixel location to a unique direction in 3D space. The focal lengths can be estimated using camera calibration techniques such as based on a chessboard pattern.

4. VISION-TRACK: ACCURATE INDOOR TRACKING

The objective is to localize the user when no anchor point is in sight. The location and the orientation obtained from the last invocation of VLC or RF based location update is used as the starting configuration. Vision-Track computes the relative displacement and relative change of orientation from this configuration to track the global coordinates of the user. For ease of exposition, we make the following assumptions:

- The user is moving in the same floor: This can be relaxed by leveraging the WiFi signature of the user. As the user moves from one floor to another, the change in the WiFi signature can be used to restart the tracking mechanism.
- The height of the camera from the floor is not changing with time: Vision-Track can be coupled with Ultrasonic depth sensing to get the depth. Alternatively, the height of ceiling can be measured using camera by tracking both the ceiling points and the floor points simultaneously. However, the only constraint is we need to know the floor-to-ceiling height (which can be known from most CAD drawings.)
- The orientation of the camera is parallel to the floor, i.e., the camera is looking at a direction parallel to the ground: This can be relaxed by incorporating the gyroscope measurements.
- The ceiling height is uniform: Later we evaluated this assumption of uniformity of ceiling height by studying the error in tracking when there is error in height of ceiling (non-uniform).

We discuss about how the last two assumptions can be relaxed in §7.

The V-axis of the camera is in the opposite direction of gravity as shown in Figure 1 and the UV axes are respectively parallel to the XY axes of the camera coordinate system. The user's trajectory is therefore traced out in the XZ plane of the ground coordinate system. Prior works such as [7], have addressed this problem by detecting complex objects such as chair, wall, cabinet, etc., followed by tracking these objects. However, implementing accurate object detection schemes for a wide range of indoor objects and maintaining an up-to-date database with their locations along with their variety of features is challenging. In sparse deployment scenarios, a user may have to move significant distances before encountering another anchor point. So, Vision-Track should accumulate minimal error with distance of movement.

Tracking Ceiling Points: Vision-Track observes the mobility of selected ceiling points across the captured video frames to measure user's movement w.r.t. ground frame of reference. Distinct points such as corners, edges etc., can be observed and tracked in a video referred to as *good features to track*. These feature points can be tracked across frames by using the *Lucas-Kanade optical flow* method [8]

with good accuracy. The OpenCV version of Android supports accurate real-time tracking of feature points.

Consider a point P_i whose position at time t in the camera plane is $(u_i(t), v_i(t))$, and in the camera coordinate system is $(x_i(t), y_i(t), z_i(t))$. The height of camera from the ground (h_a) is derived from the outcome of the earlier VLC or RF based localization. Thus, the height of the point with respect to the camera is $y_i = h_i - h_a$, where h_i is the height of P_i from the floor. From equation 1, the other two coordinates of P_i can be derived as,

$$z_i(t) = \frac{f_v(h_i - h_a)}{v_i(t)}, \qquad x_i(t) = \frac{u_i(t) f_v(h_i - h_a)}{f_u v_i(t)}. \tag{2}$$

For points on the ceiling, their height (h_c) can be obtained by communication with the anchor points that was the last one in sight or from the CAD drawing. It can also be measured by using additional sensors such as a depth sensor.

Observe that from the above equations we can obtain the location of the camera with respect to the observed point if its height from the ground is accurately known, but this is not sufficient to localize the camera, as the location of the observed point is unknown. Instead we resort to finding the instantaneous velocity so that we can integrate it to measure the relative displacement and track the motion of the camera from its previous known location.

The negative of the velocity vector of the observed point will provide the velocity vector of the user. The instantaneous velocity $(V_i(t))$ of the user in the camera coordinate system is given by:

$$V_i^z(t) = -\frac{\partial z_i(t)}{\partial t} = \frac{f_v(h_i - h_a)}{v_i(t)^2}\frac{\partial v_i(t)}{\partial t},$$
$$V_i^x(t) = -\frac{\partial x_i(t)}{\partial t} = \frac{u_i(t) f_v(h_i - h_a)}{f_u v_i(t)^2}\frac{\partial v_i(t)}{\partial t} - \frac{f_v(h_i - h_a)}{f_u v_i(t)}\frac{\partial u_i(t)}{\partial t}. \tag{3}$$

where $V_i^z(t)$, $V_i^x(t)$ are the velocities along z and x axes respectively.

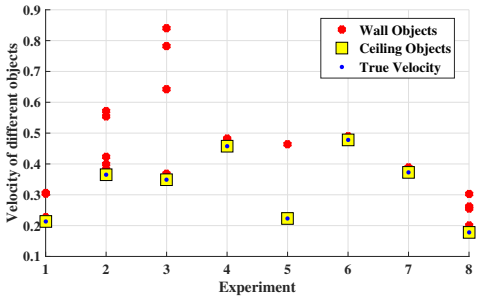

Figure 2: ***Instantaneous velocity estimated by tracking different ceiling points from an indoor walking video capture. All ceiling points are overlapping with true velocity.***

The main challenge in realizing the above idea is to identify if a point is a ceiling point or not. If a point is not on the ceiling, then its actual height is lower (as ceiling has the highest height). By mistakenly using $h_i = h_c$ during computation of $V_i^z(t)$, we end up with a larger than true value. Consider a scenario where the camera is moving forward along the z-direction without changes to its orientation. Now if we compare two points - one on the ceiling and one not on the ceiling, then the velocities along the z-axis must be the same for both. But as explained above, the measured velocity for the point not on the ceiling will be more than its correct velocity. So, the measured velocity of the ceiling point will be the lower of the two. If the camera's orientation is not changing significantly, we expect this criteria to hold. Based on this observation the *speed based algorithm* computes the speed of all the observed points and picks the one with the minimum speed. To limit the search space and to reduce false measurements, we only consider points lying in the top part of the camera frame. If multiple

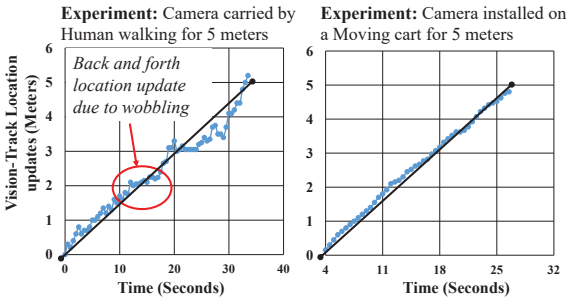

Figure 3: ***Human walking causes wobbling that cancels by itself. Black line denotes the true location***

ceiling points are observed then the ceiling points that are closer to the camera will have more distinguishable movement in the camera plane and hence, will be more robust for purposes of computing instantaneous velocity. This can also be observed from Eqn 3, where any error in $\frac{\partial v_i(t)}{\partial t}$ is magnified more significantly for smaller values of $v_i(t)$. Similarly errors in the height estimation will also get magnified more for far away ceiling points. Coincidentally, our criteria for picking up the point with the lowest velocity on the z-axis also typically favors the closest ceiling point when multiple ceiling points are in view.

Different velocity estimates and their positions such as ceiling and wall are presented in Figure 2. As shown in the figure, the point corresponding to the minimum estimate of $V_i^z(t)$ corresponds to the ceiling and also provides closest to the true velocity.

Wobbling of camera height: Human walking causes wobbling of different body points. If the camera is on a smart-glass, then it will be subjected to wobbling of the head. Wobbling causes increments and decrements to the camera height that lead to positive and negative errors that tend to cancel each other out as shown in Figure 3. This can be explained from Eqn 3, where the sign of the error (positive/negative) in height estimation determines the sign of the error in velocity estimation.

5. EXPERIMENTS

What is the localization accuracy in tracking the camera while carried by a walking human or attached to a moving cart? Vision-Track is able to track a user and

a moving cart in a corridor and office room with an error accumulation of less than 10% and 5% respectively up to 30 meters. ***What is the impact of different camera heights (height with respect to ceiling), different camera fps (frames per second) and ceiling height measurement errors?*** Vision-Track has less than a meter of error accumulation (in tracking 30 meters) for different camera heights with >20 fps and with height errors up to 10 cm, relaxing the rigid camera placement requirement. ***What is the accuracy of Vision-Track in tracking over a curving trajectory?*** Vision-Track when combined with Android gyroscope and/or geomagnetic field orientation successfully tracked users moving on curved trajectories.

Vision-Track Experimental details: *Vision-Track* is

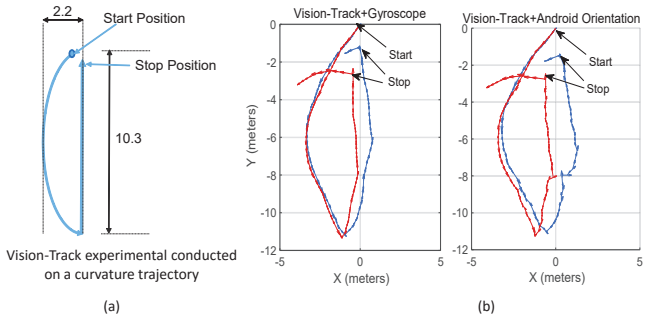

Figure 4: ***Complex trajectory tracking by Vision-Track system: (a) Shape of trajectory followed (b)Tracking D shape***

implemented in Python which uses OpenCV and fuses the gyroscope and Android orientation sensors with video processing to track the user. Gyroscope is known to lead to error accumulation which can be corrected by location and orientation estimations from the nearest anchor point. We have used iPhone 6 to capture the video and sensor readings. The obtained video and log files from sensors are merged during processing. The ground truth is measured using red colored position markers placed on the ground and an additional camera capturing the experiment for ground truth matching. Also for the following analysis camera direction is normalized to known directions so that multiple runs of experiments will be free of initial orientation errors.

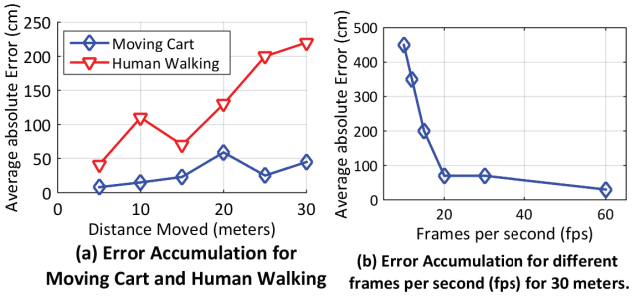

Figure 5: ***Vision-Track error: distance and fps from 30 meter tracking experiments.***

Tracking linear mobility in corridor, office room: We first evaluate error accumulation for mobility of camera along the office corridor. Figure 5(a) depicts the tracking error: (a) Camera placed on a moving cart (red line), (b) Camera carried by walking human (blue line). As shown in the figure, the tracking error accumulates slowly for case (a). Our analysis shows that non-uniformity in height of points in the ceiling can be a reason for error accumulation. For example, projecting or embedded objects in the ceiling can alter the height of such points. Also, when the ceiling objects are completely camouflaged with the indoor settings, points on the wall can be misjudged as ceiling objects. In case (b), where the camera is carried by the user, we have observed an error of 40 centimeters for every 5 meters, which is more compared to (a). This is due to two reasons: (i) *orientation error accumulation* over time for gyroscope, which is a well-studied problem [15], and, (ii) *wobbling* caused by human mobility as discussed in §4.

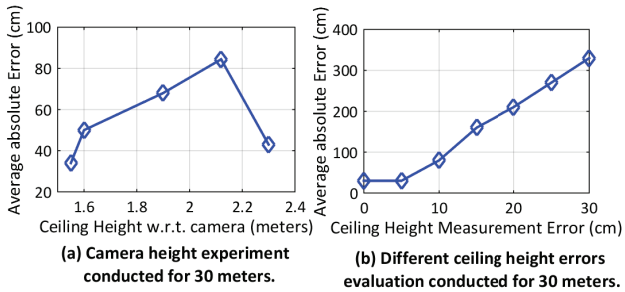

Figure 6: ***Vision-Track error: camera-to-ceiling height and camera-to-ceiling error from 30 meters tracking experiments.***

Impact of different camera heights, fps and ceiling or, camera height measurement errors: Varying heights of users lead to various possible camera mounting heights. There are many choices for the vertical placement of the smart device (e.g., head mounted or carried in hand). To study the impact of the camera mounting height, we have conducted experiments by placing camera at different heights. Figure 6(a) shows the error observed for different camera placements with respect to the ceiling. The observed error is less than a meter and varies by tens of centimeters. Figure 5(b) shows the vision-track error accumulated for tracking a distance of 30 meters. We conclude that 20 fps is sufficient to track with error accumulation of less than 5%. Figure 6(b) shows the error observed in tracking 30 meters distance when there are different ceiling height errors. As shown in the figure, the height can be relaxed to $\pm$ 5 cm to have an error of less than 35 centimeters for every 30 meters.

Tracking complex trajectories such as 'D' shape: This experiment evaluates the precision in tracking a curved trajectory. For this experiment, the video is recorded by the user moving on a 'D' shaped trajectory shown in Figure 4 (a). Figure 4 (b) shows the trajectory traced by *Vision-Track* for two experiments using (i) Gyroscope, (ii) Android orientation obtained by using API's in [1]. The initial orientation of the gyroscope is calculated using the direction of the geomagnetic field and the gravity vector. Using the orientation from these two sensors, *Vision-Track* traces the trajectory tracked by the moving camera. Both tracking error and orientation error contribute to the overall error.

Additionally, we observed that the orientation error is more when using the Android orientation sensors as observed from Figure 4(b).

6. RELATED WORK

Vision-track explores vision to track the position of the user and it is related to following works on visual odometry. **Visual Odometry:** It is the branch of vision where single or multiple mobile cameras are used to predict users' motion. It involves matching of huge number of feature points across images from multiple cameras from various locations. However, these solutions are computationally expensive as they require dense feature point matching. Additionally, single camera based visual odometry techniques (monocular visual odometry) provide the motion of camera/user by a scale factor [10] and therefore need additional information to track the user. In contrast, Vision-Track smartly identifies feature points belonging to the ceiling to measure cameras motion using a single camera. Additionally, *Vision-Track* employs feature based tracking with aided intelligence from gyroscope and inertial sensors of smart-devices and captures feature points corresponding to greatest height (ceiling points). Further, *Vision-Track* uses minimal information of the ceiling height to facilitate tracking objects using the observed feature points.

7. DISCUSSION AND FUTURE WORK

(1) Power Optimized Vision-Track: The current version of Vision-Track needs to record video continuously for tracking feature points. However, we observed that lower fps video can be used in case of a slow moving user. So, we want to develop an inertial-sensor (foot-steps/accelerometer) trigger based fps selection as a part of our future work. In the event of stationary users, Vision-Track can increase its sleep-cycle to optimize the power. *(2) Vision-Track to emulate antenna array:* The accurate tracking information can be used in different SAR based localization techniques like Ubicarse[4]. *(2) Vision-Track to track road from a moving car:* The similar idea in inverted fashion can be applied to vehicular domain by tracking the points at lowest possible height with respect to the car. The potential feature points on the road are lane markers, dividers etc. which can be tracked to improve speed estimation accuracy of vehicles. *(2) Assumption on knowledge of height:* Vision-Track assumes the height of the camera to be fixed, to track the ceiling points. However, it limits the user from free use of smart devices. This limitation can be addressed by tracking both ceiling and floor points. For example, consider equation 3, irrespective of height h_a (height of camera from floor which is unknown), the ceiling points correspond to the least possible velocity. Therefore, feature points corresponding to ceiling can be filtered out. Similarly, by observing the lower portion of the frame, one can obtain the points at lowest possible height (floor). If the orientation of the camera is assumed to be fixed (can be detected from gyroscope), the velocity of ceiling point should be same as that of floor point along Z-axis. Equating both, will solve h_a eliminating the assumption on height. However, this technique needs robust filtering to deal with noises such as human movement which happen at floor height and is left for future work.

8. CONCLUSION

Vision-Track attempts to enhance the range of RF or VLC based location services by providing continuous tracking services with no modification to the infrastructure. Vision-Track leverages the high pixel density of cameras in today's smart-phones to observe and analyze specially chosen feature points in indoor environment to accurately find the relative displacement and orientation of the smart-device carried by the user. The system evaluated in indoor college buildings demonstrates sub-meter level accuracy in tracking a user over long distances (30m).

9. ACKNOWLEDGMENTS

This work was partially supported by NSF grant CNS-1618520.

10. REFERENCES

[1] Android orientation API. http://developer.android.com/reference/android/hardware/SensorEvent.html.

[2] P. Dietz, W. Yerazunis, and D. Leigh. Very low-cost sensing and communication using bidirectional leds. In *Proc. of ACM UbiComp (2003)*, pages 175–191. Springer.

[3] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *Proc. of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282.

[4] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proc. of ACM MobiCom (2014)*, pages 483–494.

[5] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proc. of ACM MobiCom (2014)*, pages 447–458.

[6] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *Proc. of USENIX NSDI (2014)*, pages 331–343.

[7] H. Lim and S. Sinha. Towards real-time semantic localization. In *ICRA Workshop on Semantic Perception*, 2012.

[8] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[9] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. Sail: Single access point-based indoor localization. In *Proc. of ACM MobiSys (2014)*, pages 315–328.

[10] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

[11] Y. Shu, K. G. Shin, T. He, and J. Chen. Last-mile navigation using smartphones. In *Proc. of ACM MobiCom (2015)*, pages 512–524.

[12] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[13] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Proc. of NSDI (2013)*, pages 71–84.

[14] J. Xiong, K. Sundaresan, and K. Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proc. of ACM MobiCom (2015)*, pages 537–549.

[15] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *Proc. of ACM MobiCom (2014)*, pages 605–616.