Pose Invariant Activity Classification for Multi-Floor Indoor Localization

Saehoon Yi*,‡, Piotr Mirowski*†, Tin Kam Ho* and Vladimir Pavlovic‡
*Statistics and Learning Research Department, Bell Laboratories, Alcatel-Lucent
600 Mountain Avenue, Murray Hill, NJ 07974, USA
†Microsoft Bing, 100 Victoria Street, London SW1E 5JL, UK
‡Department of Computer Science, Rutgers, The State University of New Jersey
110 Frelinghuysen Road, Piscataway, NJ 08854, USA
Email: shyi@cs.rutgers.edu

***Abstract*—Smartphone based indoor localization caught massive interest of the localization community in recent years. Combining pedestrian dead reckoning obtained using the phone's inertial sensors with the GraphSLAM (Simultaneous Localization and Mapping) algorithm is one of the most effective approaches to reconstruct the entire pedestrian trajectory given a set of visited landmarks during movement. A key to GraphSLAM-based localization is the detection of reliable landmarks, which are typically identified using visual cues or via NFC tags or QR codes. Alternatively, human activity can be classified to detect organic landmarks such as visits to stairs and elevators while in movement. We provide a novel human activity classification framework that is invariant to the pose of the smartphone. Pose invariant features allow robust observation no matter how a user puts the phone in the pocket. In addition, activity classification obtained by an SVM (Support Vector Machine) is used in a Bayesian framework with an HMM (Hidden Markov Model) that improves the activity inference based on temporal smoothness. Furthermore, the HMM jointly infers activity and floor information, thus providing multi-floor indoor localization. Our experiments show that the proposed framework detects landmarks accurately and enables multi-floor indoor localization from the pocket using GraphSLAM.**

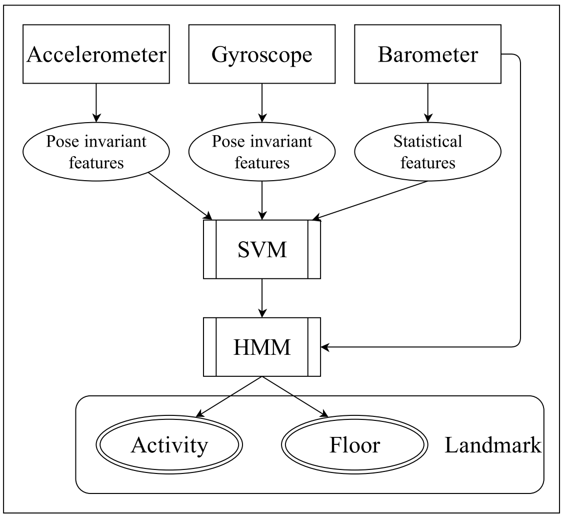


Fig. 1: General framework

I. Introduction

Indoor geo-localization became prominent thanks to the widespread use of smartphones, which contain various built-in sensors such as accelerometer, gyroscope, barometer, not to mention radio frequency sensor for WiFi or cellular networks. Data collected by these sensors enable new ways for the user to localize herself using a single self-contained device. Among the many applications of indoor localization are: helping users to navigate within a building in real time or geo-targeted advertisement that allows stores in a shopping mall to send location-based advertisement to consumers nearby. For network planners and operators, localization of users can help identify areas with weak cellular network signals in order to suggest placements of small cells to provide flawless connectivity.

Localization can be achieved using various methods. One popular approach is radio fingerprint mapping [1], [2], where the current location is inferred by matching similar signals in the fingerprint map. The accuracy depends on the quality of the map, which is often expensive to acquire with high precision and accuracy. As an alternative, a dead-reckoning system provides localization with less expense, but it occasionally requires additional data processing for accurate localization. For instance, Pedestrian Dead-Reckoning (PDR) estimates a trajectory by incremental updates thanks to a step detector and compass readings. The current location is calculated by translating from the previous location by a step size along the movement direction. However, the error is accumulated over time and it may lead to considerable drift.

The GraphSLAM algorithm [3] effectively estimates an accurate PDR trajectory from raw traces that contain drift error. When a pedestrian visits a particular landmark more than once, a PDR-calculated trajectory may yield different positions at every visit because of the accumulated errors. Provided data association between landmarks, "loop closure" can be solved and the GraphSLAM minimizes this discrepancy along the trajectory. Therefore, in order to obtain a good optimization result, the algorithm requires a suitable number of re-visited landmarks. Landmarks can be obtained by visual cues from a camera, which requires to train for a classification of the scene surrounding the landmark. They can also be obtained by having a smartphone reading a QR code or an NFC tag registered with the landmark [4]. In the latter, the tags contain geolocation information and they have to be installed in the building beforehand. Alternatively, pedestrian activity can be classified

Preprint submitted to 22nd International Conference on Pattern Recognition.
Received December 20, 2013.

to detect visits to stairs and elevators from smartphone sensor data. Gusenbauer et al. [5] classifies activities using an SVM (Support Vector Machine) on statistical features extracted from accelerometer in a smartphone. Hardegger et al. [6] introduces a set of rule-based conditions on accelerometer readings to detect activities.

In this paper, we propose a novel activity classification system for multi-floor indoor localization. Our contributions are as follows:

- We extended the design of pose invariant features for an activity classification task. Pose is defined by how a person puts a smartphone in the pocket. We show that pose invariant features can be used to successfully classify activities.
- We designed a Hidden Markov Model that enables the integration of activity classification and floor inference.
- We applied the GraphSLAM algorithm with our activity and floor detection framework to provide multi-floor localization in a building.

The remainder of this paper is organized as follows. Section II explains pose invariant feature and statistical features extracted within our framework. Our proposed framework is described in Section III. Section IV introduces multi-floor extension of GraphSLAM. Quantitative and qualitative analysis of the experiment results are evaluated in Section V. Merits of the proposed framework are discussed in Section VI, followed by concluding remarks in Section VII.

II. Feature Extraction

A. Pose Invariant Feature for IMU Sensors

Modern smartphones contain various sensors for versatile purposes. Inertial measurement units (IMU) such as accelerometers and gyroscopes detect linear and angular acceleration which describes the motion of the smartphone. Data from both sensors are represented as 3D time series of positions with respect to x, y, and z axes which are aligned with the smartphone. As a result, such sensor readings depend on the pose of the smartphone, which is defined as the orientation of the phone in the pocket. A pose-invariant system is strongly desirable because it frees the user from the restriction of keeping the smartphone in a particular orientation.

As illustrated in Figure 2, acceleration data for the same motion are distinct when the phone is in two different poses. Rotation of the smartphone pose changes magnitudes of x, y and z values of the accelerometer data. It changes not only the average magnitudes, but also the temporal dynamics and the time of the peak. Therefore, standard features of mean and moments that [7] and [5] use will also be distinctive for different poses. A pose invariant classifier can be obtained either by collecting training samples in all poses [7] or by extracting rotation invariant features from the raw data [8].

Kobayashi et al. [8] identified that the autocorrelation of acceleration data is invariant to the rotation of the accelerometer:

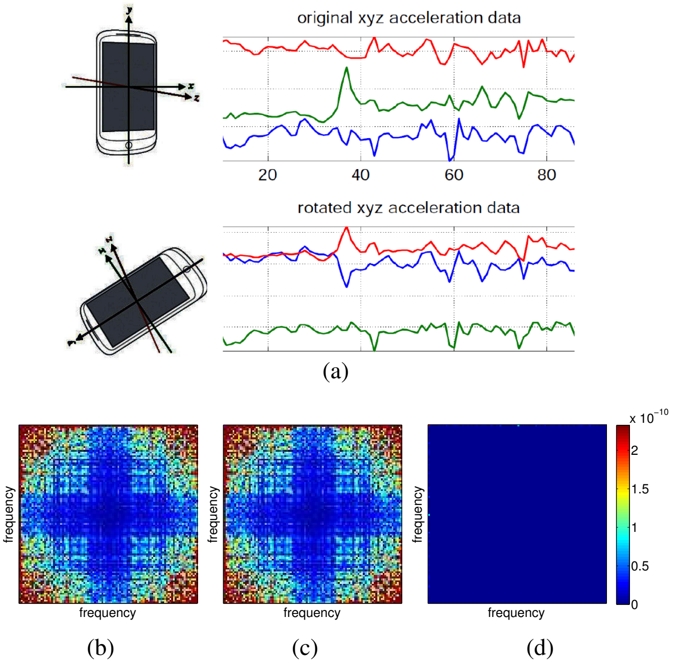


Fig. 2: Example of the pose invariant feature for accelerometer data. (a) Raw accelerometer readings on the rotation.[1] (b) Pose invariant feature extracted from the original data. (c) Pose invariant feature extracted from the rotated data. (d) The difference between (b) and (c).

$$f(\omega) = \int exp(-i\omega t)s(t)dt \in \mathbb{C}^3, \tag{1}$$

$$F = [f(\omega_1), f(\omega_2), \ldots, f(\omega_n)] \in \mathbb{C}^{3\times \mathrm{n}}, \tag{2}$$

$$A = F^* F. \tag{3}$$

3D signal $s(t) \in \mathbb{R}^3$ is obtained from accelerometer. Signal $f(\omega)$ is calculated by Fourier transform on $s(t)$. Autocorrelation matrix A in the frequency domain is acquired by multiplying the conjugate transpose of the Fourier transformed signal. A is a n × n symmetric matrix where n is the number of frequencies.

The pose invariant property is inherited from the fact that the rotation matrix R is an orthogonal matrix, $R^T R = I$. In Eq. (4), $\hat{s}$ is the rotated signal from a different pose of the phone. The rotation matrix $R = R_x R_y R_z$, where R_x, R_y and R_z are the rotation matrices with respect to each axis x, y and z.

$$\hat{s}(t) = Rs(t). \tag{4}$$

Since the Fourier transform is a linear operation, we have the following property:

[1] Portions of this figure are modifications based on work created and shared by the Android Open Source Project and used according to terms described in the Creative Commons 2.5 Attribution License.

$$\begin{aligned}\hat{f}(\omega) &= \int exp(-i\omega t)\hat{s}(t)dt, \\ &= R\int exp(-i\omega t)s(t)dt, \\ &= Rf(\omega). \end{aligned} \tag{5}$$

Eq. (5) becomes $\hat{F} = RF$ in matrix form. Eventually, the autocorrelation matrix A is rotation invariant as in Eq. (6). Therefore, the feature A provides a pose invariant property.

$$\hat{A} = \hat{F}^*\hat{F} = F^*R^TRF = F^*F = A. \tag{6}$$

In Figure 2, the pose invariant feature extracted from the rotated accelerometer data is nearly identical to the original data. The difference is bounded to 10^{-10} in the magnitude of correlation strength due to a precision error.

We extend the idea and apply the pose invariant features on both accelerometer and gyroscope sensor data to classify pedestrian activity. Autocorrelation matrix is estimated for each sliding window of the data and the upper triangle components of the autocorrelation matrix is used as the pose invariant features. Kobayashi et al. [8] showed that the proposed features of accelerometer readings were able to identify a person by his gait. Our experiments show that the features are successful in classifying different activities when trained on activity labels.

B. *Statistical Features from a Barometer*

For locomotive activities of walking and taking stairs, inertial sensors provide enough information to classify the activities. However, for the activity of taking elevators, the inertial sensors observe zero acceleration inside of a moving elevator. Thereby, Alzantot and Youssef [9] detect elevator activities by a state model, which catches unique acceleration patterns when the elevator starts to move and stops. However, we want to integrate the elevator classifier with other activities in the same sliding window framework. Therefore, the only information that helps detecting the elevator movement is the air pressure from a barometer sensor. Air pressure changes as the elevator moves up and down. However, barometer readings consistently fluctuate even if the sensor stays at the same level, thus we need to use some statistical features to get robust observations, as listed in Table I.

It is worth to note that we take the sliding window of a small size to ensure short latency of activity detection. Therefore, the barometric pattern of a stair-climbing activity is hardly distinguishable from that of a walking activity on the same floor. As a consequence, barometric features should be used together with inertial sensor data for classification of all activities.

III. Proposed Framework

In this section, we propose a novel Bayesian framework for integrating activity classification from a Support Vector Machine(SVM) with floor inference using a Hidden Markov Model(HMM). The overview of the framework is depicted in Figure 1.

TABLE I: Statistical features for barometer data.

Relative to initial point	$\tilde{b}(t) = b(t) - b(1)$
Velocity	$v(t) = b(t) - b(t-1)$
Acceleration	$a(t) = v(t) - v(t-1)$
Mean	$\mu = \frac{1}{n}\sum_{t=1}^{n}\tilde{b}(t)$
Mean of 1^{st} half	$\mu_1 = \frac{2}{n/2}\sum_{t=1}^{n}\tilde{b}(t)$
Mean of 2^{nd} half	$\mu_2 = \frac{2}{n}\sum_{t=n/2+1}^{n}\tilde{b}(t)$
Difference of means	$\mu_\delta = \mu_2 - \mu_1$
Slope	$\theta = b(n) - b(1)$
Variance	$\sigma^2 = \frac{1}{n}\sum_{t=1}^{n}(\tilde{b}(t) - \mu)^2$
Standard deviation	$\sigma = \sqrt{(}\sigma^2)$
Root mean square	$r = \sqrt{(}\frac{1}{n}\sum_{t=1}^{n}\tilde{b}(t)^2)$
Signal magnitude area	$s = \frac{1}{n}\sum_{t=1}^{n}\lvert\tilde{b}(t)\rvert$

A. *SVM Activity Classification*

In our framework, features are extracted from smartphone sensor data and passed to an SVM for activity classification. As described in Figure 1, rotation invariant features are extracted from accelerometer and gyroscope sensors and statistical features from barometer in each sliding window of an input sequence. A linear SVM model classifies each sliding window sample and generates class probability from Platt's scaling algorithm [10].

SVM classification is limited to observations from one sliding window and has no ability to maintain reference to activities occurring in previous sliding windows. Hence, there may arise sporadic misclassifications. Classification results can be improved if we promote temporal smoothness on the activity sequence.

B. *HMM Activity and Floor Inference*

HMM is one of the most popular time series models and it captures temporal dependency among states. Activity classification results obtained from the SVM can be refined by an HMM if we define activities as states and suppress the unlikely state transitions. Furthermore, by extending the definition of a state as a joint identification of the activity and the floor, state inference can integrate activity with floor inferences. Such a combined state will help constrain the state transition. For instance, one can transit from one floor to the consecutive floors only by the activities of taking stairs and elevators. Given this definition of a state, the proposed HMM consists of transition and observation probabilities that are explained as follows.

Transition probability Traditionally, transition probabilities of hidden states are learned from training data by the Baum-Welch algorithm. Unlike the traditional HMM, we explicitly define the states, and data-driven hidden states may not match our defined states. Therefore, motivated by Li et al. [11], we manually design the transition probabilities as shown in Figure 3. It results from the fact that activity transition occurs sparsely over time, thus probability of state transition is much lower than staying in the same state. Moreover, the transition between certain activities is not possible. For example, between elevator and walking activities, a pedestrian should stand still

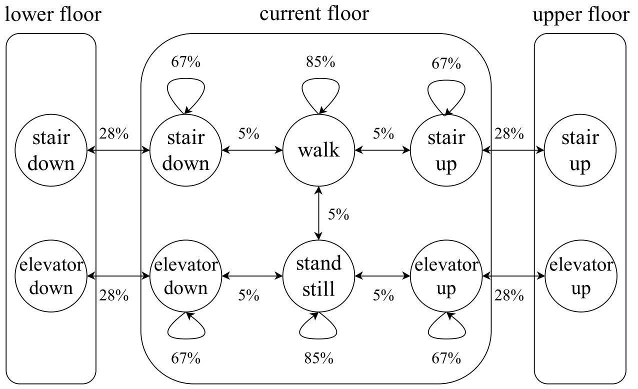


Fig. 3: State transition diagram for the HMM.

for a moment. Therefore, the state of elevator activity first changes to standing still before changing to walking.

Observation probability Observation probabilities are obtained jointly from activity and floor likelihood. Air pressure observation from a barometer y_{floor} is modeled by a mixture of Gaussians, where each floor forms a Gaussian distribution with $(\mu_{\text{floor}}, \sigma_{\text{floor}})$. As the altitude increases, the air pressure decreases. Therefore, μ_{floor} of the air pressure decreases linearly for the higher floors. $p(s_{\text{floor}_i}|y_{\text{floor}})$ is the floor posterior from the mixture of Gaussian model. Activity class posterior $p(s_{\text{act}_i}|y_{\text{act}})$ is estimated from Platt's scaling on SVM decision values as described in Section III-A. Activity and floor are independent events because each activity can occur independently on every floor, so the joint probability is calculated as Eq. (8). T is the sequence length and S is the number of states in Eq. (9).

$$p(y|s_i) = \frac{p(s_i|y)p(y)}{p(s_i)}, \tag{7}$$

$$p(s_i|y) = p(s_{\text{floor}_i}|y_{\text{floor}})p(s_{\text{act}_i}|y_{\text{act}}), \tag{8}$$

$$p(y) = \frac{1}{|T|}, p(s_i) = \frac{1}{|S|}. \tag{9}$$

Once the observation probabilities are estimated, the most probable state sequence is inferred by the Viterbi algorithm. As a result, each HMM state is decomposed into activity and floor labels.

C. Post-Process Rectification

The HMM smooths the state transition because the probability of state change is much smaller than that of staying in the same state. Thus, the number of sporadic misclassifications from the SVM may be reduced. In addition, activity inference of the HMM can be further improved by rectifying activities of *stairs* that involve no floor change to *walk* and, likewise, *elevators* to *stand still*.

IV. Multi-floor GraphSLAM with Organic Landmark

GraphSLAM is an approach that optimizes a trajectory by representing it as a graph of constraints between consecutive positions and by minimizing an error to satisfy the constraints specified by the graph. Each node in the graph corresponds to the location x of a pedestrian obtained from pedestrian dead reckoning (PDR). An edge represents a constraint between the nodes. As the pedestrian moves, consecutive nodes of the location are connected by an edge z (i.e., consecutive steps). GraphSLAM minimizes the error $e = z - \hat{z}$, the difference between the real constraint and the observed constraint. When a person re-visits a landmark, an additional edge is added from the current node to the node that corresponds to the same landmark which was visited at an earlier time. Such constraint edges z are assumed to have a zero weight because they connect a location with itself (loop closure). GraphSLAM expresses the error as a sum of quadratic constraint at each edge and optimizes the total error by iteratively linearizing the error and solving a linear least squares problem until convergence. In order to obtain an accurate trajectory, GraphSLAM requires a good number of landmarks visited more than once. Detailed explanation and formulation can be found in the tutorial [3].

In this paper, we focus on providing organic landmarks which are stairs and elevators detected when a pedestrian moves inside a building. The identity of landmarks can be determined by comparing WiFi visibility signatures such as the MAC address of a WiFi access point. On training, WiFi visibility and the physical location of all landmarks are obtained as a reference landmark list. WiFi visibility of a landmark is represented by a binary vector, which is filled with 1 when a WiFi access point is visible from the landmark and 0 otherwise. This binary vector is rescaled and normalized to be an estimate of probability density function of access points. The χ^2 distance is employed as a distance metric between two probability density functions from different landmarks. Therefore, when a landmark is detected on testing, we compare the current WiFi visibility to all landmarks and take the physical location of the closest landmark in the reference list.

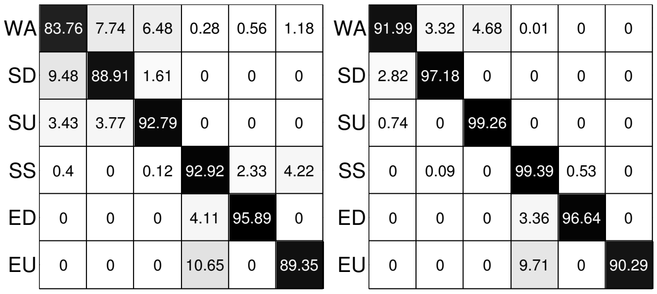
Unlike the sit and walk ActionSLAM landmarks [6], stair and elevator landmarks are appropriate in multi-floor environments and they help reduce the need for manual landmark installation, such as NFC tags or QR codes which may be laborious to maintain [4]. In addition, our framework infers floor information along the trajectory. Integrating trajectory with floor information gives a multi-floor trajectory inside a building.

V. Experiments and Results

We experimented with the proposed method in a large, multi-floor office building with many stairs and elevators. In our experiments, accelerometer, gyroscope and barometer data are recorded from Android smartphones at 50Hz. Training data were recorded for a total of 10271 seconds performed by three subjects. To help with annotation, the same action was performed repeatedly. We defined 6 indoor activities of *walking, taking stairs down, taking stairs up, standing still, taking elevator down* and *taking elevator up*. For test data, subjects walked inside a building naturally. Test trajectories are composed of 12 sequences in total of 6160 seconds long. The ground truth of the activity and floor information was manually recorded for the purpose of validating the results. An SVM takes pose invariant features from accelerometer and gyroscope data in sliding windows of size 64 time points (corresponding

TABLE II: Activity classification accuracy (%).

Model	SVM	HMM	Rect
Walk (WA)	83.76	91.99	99.22
Stair Down (SD)	88.91	97.18	97.47
Stair Up (SU)	92.79	99.26	99.32
Stand Still (SS)	92.92	99.39	100.00
Elev. Down (ED)	95.89	96.64	96.64
Elev. Up (EU)	89.35	90.29	91.34
All	87.45	94.82	99.25



	WA	SD	SU	SS	ED	EU
WA	83.76	7.74	6.48	0.28	0.56	1.18
SD	9.48	88.91	1.61	0	0	0
SU	3.43	3.77	92.79	0	0	0
SS	0.4	0	0.12	92.92	2.33	4.22
ED	0	0	0	4.11	95.89	0
EU	0	0	0	10.65	0	89.35

(a) SVM

	WA	SD	SU	SS	ED	EU
WA	91.99	3.32	4.68	0.01	0	0
SD	2.82	97.18	0	0	0	0
SU	0.74	0	99.26	0	0	0
SS	0	0.09	0	99.39	0.53	0
ED	0	0	0	3.36	96.64	0
EU	0	0	0	9.71	0	90.29

(b) HMM

Fig. 4: Activity confusion matrices (%).

to 1.28 seconds) and of step size 35. The energy in the 64 frequencies are obtained by Fourier transform and the upper triangle matrix of the autocorrelation matrix is used as the pose invariant features. Statistical features from barometer data are extracted from sliding windows of size 192 time points which triples the window size of the inertial sensor data. It is so to help capture increments or decrements of the air pressure within a sliding window.

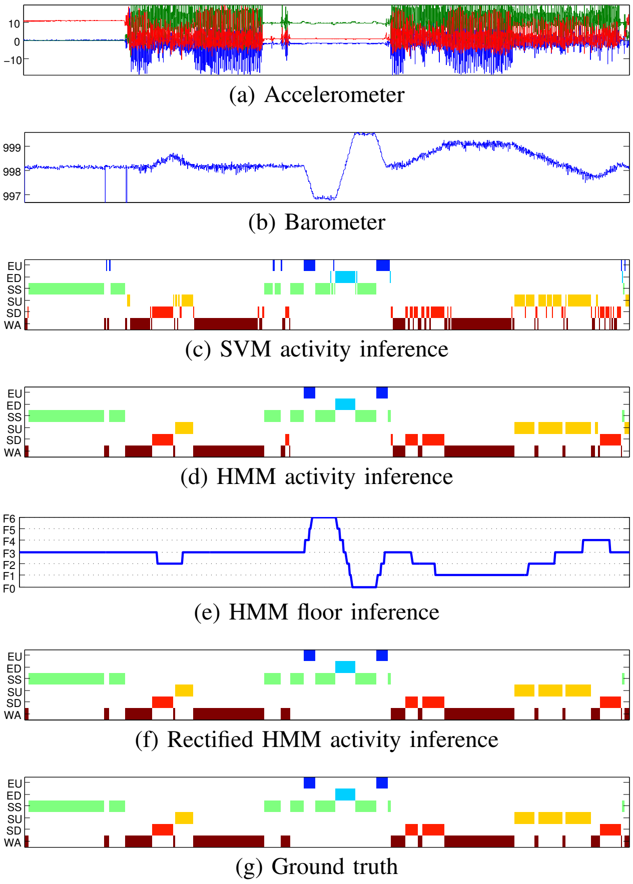
A. *Quantitative analysis*

Activity classification results for various models are shown in Table II. Columns show class accuracies for SVM(Section III-A), HMM(Section III-B) and rectification results (Section III-C), respectively. The HMM inference obtained from the Viterbi algorithm improves over the SVM classification for all activities. Figure 4 shows that the HMM improves confusions on locomotive activities of *walk, stair down* and *stair up*. It also improves misclassifications of the activities of *stand still* to *elevator down* and *elevator up*. Such sporadic misclassifications were suppressed by temporal smoothing from the HMM. Finally, post-processing with HMM inference further rectifies the *walk* activity which was misclassified as *stairs*.

B. *Qualitative analysis*

Figure 5 shows an example of the inference result. Figure 5c, 5d and 5f illustrate activity inference over time. The labels of activities are *walking(WA), stair down(SD), stair up(SU), stand still(SS), elevator down(ED)* and *elevator up(EU)* from bottom to top. Figure 5e depicts floor inference over time. In the given sequence, the user visited 7 floors including $F0$ which is the basement.

We observe that SVM inference in Figure 5c gives misclassification between locomotive activities of *walking, stair down* and *stair up*. Those misclassifications are corrected by the HMM Viterbi algorithm as in Figure 5d. The floor is correctly inferred by the Viterbi algorithm as in Figure 5e. Post-processing in Figure 5f further corrects *stairs down* and *stairs up* activities that did not incur a floor change to *walk*.



(a) Accelerometer

(b) Barometer

(c) SVM activity inference

(d) HMM activity inference

(e) HMM floor inference

(f) Rectified HMM activity inference

(g) Ground truth

Fig. 5: Activity and floor inference result.

C. *Multi-floor GraphSLAM*

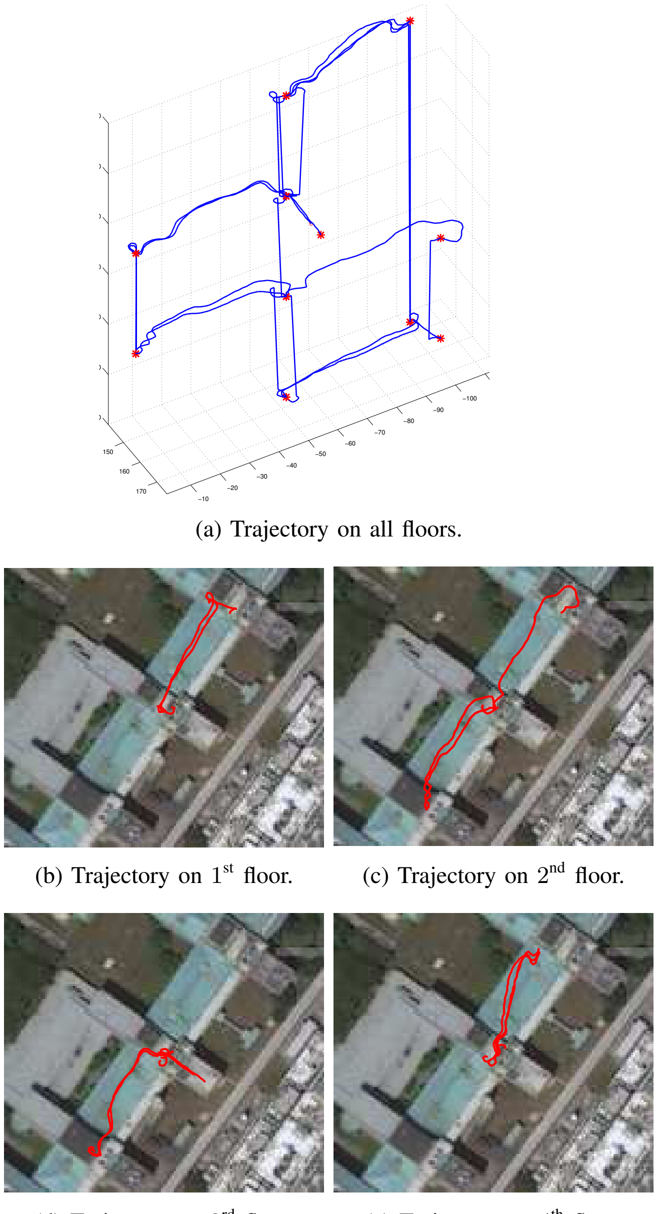
Once our framework detects stairs and elevators, we provide the detection results as organic landmarks. When the activity classifier detects a landmark of stairs or elevator, the identity of the landmark is determined by visibility signatures of WiFi access point as described in Section IV.

Multi-floor localization result is shown in Figure 6. First, a 2D GraphSLAM algorithm [3] optimizes a raw trajectory acquired by pedestrian dead reckoning (PDR) with organic landmarks. Then, the rectified trajectory is integrated with the floor inference. Trajectory on all floors are shown in Figure 6a. Red markers represent landmarks. This example describes a trajectory that included visits to 4 floors. The trajectory on each floor is shown in Figure 6b, 6c and 6d.

VI. Discussion

The merits of the proposed framework are two-fold.

First, the pose invariant feature enables efficient learning for the SVM classifier. Albert et al. [7] achieved a pose invariant activity recognition model by collecting accelerometer



(a) Trajectory on all floors.

(b) Trajectory on 1st floor.

(c) Trajectory on 2nd floor.

(d) Trajectory on 3rd floor.

(e) Trajectory on 4th floor.

Fig. 6: Multi-floor GraphSLAM result.

data in all poses. They defined 4 poses in *(a) screen in/right side up, (b) screen in/upside down, (c) screen out /right side up and (d) screen out/upside down*. Our proposal is more efficient for the following reasons: A user may put a phone in the pocket in a pose other than the listed poses. Our use of pose invariant feature enables robust classification on arbitrary poses. In addition, it avoids the laborious process to collect and annotate the data for each activity in all poses.

Second, the HMM module within our framework effectively integrates together the activity and floor inference. Previous studies either recognized human activity from inertial motion unit sensors [5], [7], or inferred the floors from barometric sensors [12], [13]. To our best knowledge, the activity and floor recognition have not been considered at the same time. In contrast, the proposed framework jointly infers activity and floor information. We show in the Table II that the integration increases activity classification accuracy because the activity and floor estimates provide complementary information.

VII. Conclusion

In this paper, we propose a novel framework that jointly infers activity and floor landmarks. Pose invariant features from inertial sensors are adopted for SVM-based activity classification. We design an HMM where an activity on each floor defines a state. State transitions are designed to provide temporal smoothness of a state sequence. The probability of an observation is estimated from an activity class probability provided by the SVM classification, which is multiplied by the floor likelihood from a mixture-of-Gaussians model. We further rectify activity inferences when we observe activities of stairs and elevators which do not incur floor changes. Our experiments show that the proposed framework accurately classifies activities and infers floors. Finally, we showed that the organic landmarks obtained from our framework can be applied effectively to enable multi-floor GraphSLAM.

References

[1] P. Bahl and V. Padmanabhan, "An in-building RF-based user location and tracking system," in *Proceedings of IEEE Infocom*, vol. 2, 2000, pp. 775–784.

[2] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless lans," *Comput. Netw.*, vol. 47, no. 6, pp. 825–845, Apr. 2005.

[3] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[4] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald, "SignalSLAM: Simultaneous localization and mapping with mixed wifi, bluetooth, lte and magnetic signals," in *International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 361–370.

[5] D. Gusenbauer, C. Isert, and J. Krsche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *International Conference on Indoor Positioning and Indoor Navigation*, 2010.

[6] M. Hardegger, S. Mazilu, D. Caraci, F. Hess, D. Roggen, and G. Tröster, "ActionSLAM on a smartphone: At-home tracking with a fully wearable system," in *International Conference on Indoor Positioning and Indoor Navigation*, 2013.

[7] M. V. Albert, S. Toledo, M. Shapiro, and K. Koerding, "Using mobile phones for activity recognition in parkinson's patients," *Frontiers in Neurology*, vol. 3, no. 158, 2012.

[8] T. Kobayashi, K. Hasida, and N. Otsu, "Rotation invariant feature extraction from 3-D acceleration signals," in *ICASSP*, 2011, pp. 3684–3687.

[9] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," *CoRR*, vol. abs/1209.3794, 2012.

[10] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[11] A. Li, L. Ji, S. Wang, and J. Wu, "Physical activity classification using a single triaxial accelerometer based on HMM," in *International Conference on Wireless Sensor Network, 2010*, 2010, pp. 155–160.

[12] H. Wang, H. Lenz, A. Szabo, U. D. Hanebeck, and J. Bamberger, "Fusion of barometric sensors, wlan signals and building information for 3-D indoor/campus localization," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 426–432.

[13] A. G. Ozkil, Z. Fan, J. Xiao, S. Dawids, J. Kristensen, and K. H. Christensen, "Mapping of multi-floor buildings: A barometric approach," in *Intelligent Robots and Systems, 2011*, 2011, pp. 847–852.