Smartphone-based User Location Tracking in Indoor Environment

Team Members: Viet-Cuong Ta\textsuperscript{1,2}, Dominique Vaufreydaz\textsuperscript{1}, Trung-Kien Dao\textsuperscript{1}, Eric Castelli\textsuperscript{2}

\textsuperscript{1} Pervasive Interaction/LIG, CNRS, University of Grenoble-Alpes, Inria, France
\textsuperscript{2} MICA Institute (HUST-CNRS/UMI2954-Grenoble INP), Hanoi University of Science and Technology, Vietnam
The whole path is split into:
- Find Building ID
- Find Floor ID
- Path Approximation
- Smoothing
- What next?

Figure 1: Subtasks and sensors are used
Use GNSS is enough.

If not, we can look into the BSSID of the WIFI.

<table>
<thead>
<tr>
<th></th>
<th>UAH</th>
<th>CAR</th>
<th>UJITI</th>
<th>UJIUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAH</td>
<td>0.</td>
<td>24.9</td>
<td>285.0</td>
<td>284.6</td>
</tr>
<tr>
<td>CAR</td>
<td>24.9</td>
<td>0.</td>
<td>292.7</td>
<td>293.1</td>
</tr>
<tr>
<td>UJITI</td>
<td>285.0</td>
<td>292.7</td>
<td>0.</td>
<td>0.4</td>
</tr>
<tr>
<td>UJIUB</td>
<td>284.6</td>
<td>293.1</td>
<td>0.4</td>
<td>0.</td>
</tr>
</tbody>
</table>

**Table 1:** Distance between buildings in km.
Floor Identification

- Use WIFI data, by finger-printing approach.
  - Group “closed” WIFI data into one complete scan
  - Sparse data
- Feature set:
  - Raw feature: $D = 353$ in case of UAH building
  - K-filter feature$^{[1]}$: used $K = 2$
  - Hyperbolic Location Features (HLF)$^{[2]}$


Floor Identification

- Learning models: KNN, Random Forest (RF), Extreme Gradient Boosting (XGB)\[^{[3]}\]

- Results in cross-validation testing, with 5-fold:

- End up with two assumptions:
  - Floor is well-separated
  - Entrance/leaving points are at the stairs

<table>
<thead>
<tr>
<th></th>
<th>RAW</th>
<th>2-filters</th>
<th>HLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>91.47%</td>
<td>91.30%</td>
<td>91.47%</td>
</tr>
<tr>
<td>RF</td>
<td>95.52%</td>
<td>94.28%</td>
<td>92.70%</td>
</tr>
<tr>
<td>XGB</td>
<td>98.24%</td>
<td>97.80%</td>
<td>97.36%</td>
</tr>
</tbody>
</table>

Path Approximation within a Floor: WIFI

- Use WIFI fingerprinting approach:
  - The same feature set and learning models as floor.
  - Change the target: regression and classification.
- An average of error at 3rd-quarter is around 6.5m with cross validation

<table>
<thead>
<tr>
<th>Method</th>
<th>Raw</th>
<th>2-filters</th>
<th>HLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN regression</td>
<td>9.7m</td>
<td>9.4m</td>
<td>9.1m</td>
</tr>
<tr>
<td>KNN classifier</td>
<td>10.3m</td>
<td>10.3m</td>
<td>10.3m</td>
</tr>
<tr>
<td>RF classifier</td>
<td>10.6m</td>
<td>11.5m</td>
<td>12.9m</td>
</tr>
<tr>
<td>XGB classifier</td>
<td>6.6m</td>
<td><strong>6.0m</strong></td>
<td>6.2m</td>
</tr>
</tbody>
</table>

**Table 3:** 3rd-quarter error of several learning models
Path Approximation within a Floor: Speed

- For speed:
  - Moving and standing patterns are well separated.
  - From the log file, calculate the average speed.
- Use simple rule:
  - If \( \text{std} \geq 1.0 \), use average speed
  - Otherwise, 0

Figure 2: Z-axis of accelerometers
Path Approximation within a Floor: Direction

- Direction is calculated in a numerous way:
  - By rotation matrix from ACCE and MAGN
  - By integrating of GYRO data
  - By Madgwick filter\[^4\]
  - By AHRS data
- The path is constructed by using Particle Filter\[^5\]
- Affect by errors drifting seriously

Figure 3: Four different methods for computing direction

\[^4\] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK), 2010
Path Approximation within a Floor: Combination

- Combining with WIFI
  - It takes around 4s to get a new completed WIFI scan
  - Use local adjustment from the classifier results

Figure 4: Adjusting particle P based on output of WIFI fingerprinting classification model
Path Approximation within a Floor: Wall-crossing check (1)

- Wall crossing adjustment:
  - Assign the direction to go parallel with the wall

*Figure 5*: Avoiding to cross the wall by adjusting the local direction
Path Approximation within a Floor: Wall-crossing check (2)

- Optimizing wall crossing:
  - Use 2 operators: rotation and local speed adjust.
  - Greedy algorithm: apply to avoid first cross wall.
  - Can be solved by dynamic programming but difficult

Figure 5: Results of applying greedy algorithm for adjusting speed and direction
Path Approximation within a Floor: Results

Results on 3 minutes and 7 minutes approximation:

<table>
<thead>
<tr>
<th>Method</th>
<th>3 minutes</th>
<th>7 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIFI</td>
<td>16.4m</td>
<td>29.8m</td>
</tr>
<tr>
<td>Wall adjust + WIFI</td>
<td>14.2m</td>
<td>28.1m</td>
</tr>
<tr>
<td>Optimize + Wall adjust + WIFI</td>
<td>10.1m</td>
<td>24.5m</td>
</tr>
</tbody>
</table>

Table 4: 3rd-quarter error of three combining methods
Path Approximation within a Floor: Results

- Best approximation results (after submitting the paper):
  - Use rotation matrix only with normalization to 0-mean MAGN (from our paper’s reviewers).
  - Only use local adjust with WIFI
  - Do forward and reverse approximation then take weighted average position.
  - Error 3rd-quarter is around 13.0m

Figure 6: Best approximating results on Floor 1, Route 1, S3 phone, UAH building.
Discussion and Future Works

- The test data is the real challenge.
- The problem is not solved yet:
  - Floor is not well separated enough on test data: cannot identify entrance/leaving points.
    - Proposed solutions:
      - Moving patterns can be used here (turning around in the stairs/standing in elevators)
      - Depend largely on WIFI at first step
      - Looking for big changes in MAGN
  - If the phone is in the pocket? Proposed solutions:
    - Use WIFI only.
    - Use other axis, however when?
THANK YOU FOR LISTENING!
and Wish You Have A Good Accuracy