Northeastern University

Inferring User Routes and Locations using Zero-Permission Mobile Sensors

Sashank Narain, Triet D. Vo-Huu, Kenneth Block and Guevara Noubir
College of Computer and Information Science
Northeastern University, Boston, MA

Motivation

- Leakage of location information a major privacy concern
  - Can be used to track users, find their identity or home / work locations
- Mobile OSs have some protections to prevent location access
  - Permissions for accessing location information
  - Increasing awareness among users regarding location privacy
    - But many still careless (E.g. 4.7 stars for Brightest flashlight app)
- Protecting location leakage from side-channels a harder problem
  - No permissions for accessing sensors or restrictions on rate
  - No notifications to users about access

[FTC Approves Final Order Settling Charges Against Flashlight App Creator]

Goal: Demonstrate feasibility of using smartphone sensors to infer user routes with high probability

Outline

- Graph Theoretic Approach
- Map Data Graph Construction
- Sensors for Inference
- Sensor Data Route Construction
- The Search Algorithm
- Evaluation Results (simulation and real)

Graph Theoretic Approach

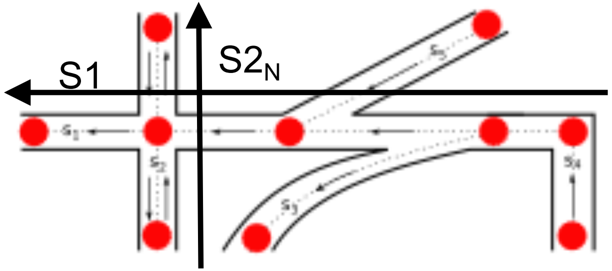
- Preparation (One-time)
  - Download road network for areas
  - Convert information to graph G = (V, E)
- Data Collection
  - Detect and record sensor data of user driving
- Data Processing
  - Perform noise correction and alignment
  - Convert aligned data to subgraph
- Search
  - Search maximum likelihood route on graph

Block diagram of the attack

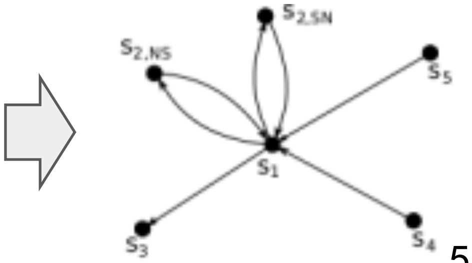
Map Data Graph Construction

- Extract map data
  - Road information from OpenStreetMaps & Speed limits from Nokia HERE platform
- Construct directed graph
  - Decompose each road into one-way atomic sections
    - Sections - road between two intersections / end-points
    - Does not contain turns or sharp curves
    - Contains curve, heading and minimum time (from speed limit + overspeed)
  - Reconstruct atomic sections to form segments
    - Segments - Many sections connected to form straight or curved road

Example Road Network



Generated Graph

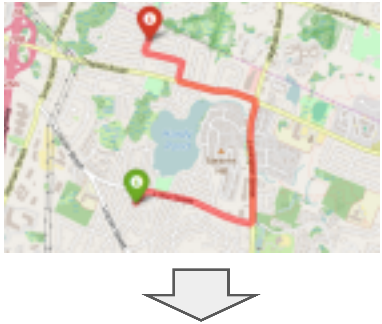


Sensor Data

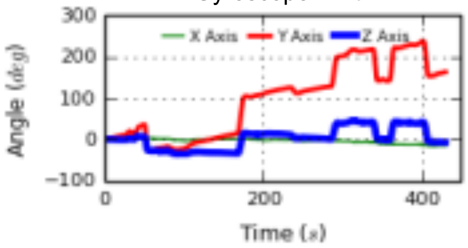
- Gyroscope
  - Extract turn angles and curvature
  - ***Most stable and useful for inference***
- Accelerometer
  - Calculate idle time
- Magnetometer
  - Calculate heading direction

Sensor Limitations

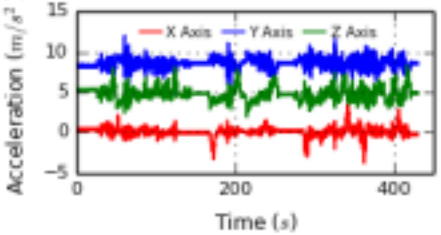
- Gyroscopes drift
  - Values drift away from axis (axis misalignment)
- Accelerometers not suited for speed estimation
  - Extremely sensitive to motion and very noisy
  - Vibrations, potholes, road slopes induce large accelerations
  - Difficult to remove bias (user calibration required)
- Magnetometers add difficulty in heading estimation
  - Extremely sensitive to car electromagnets (fans, speakers)



Gyroscope Drift



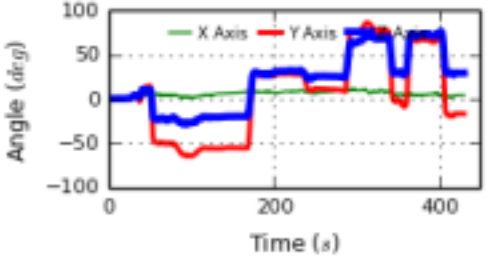
Accelerometer Noise



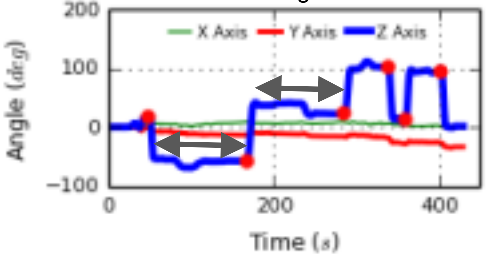
Sensor Data Route Construction

- Reduce drift from Gyroscope data
- Align to horizontal reference frame
    - Puts turn information in z axis
- Detect turns (edges) and extract segment (vertices)
    - Segment - Trace between two turns (includes curvature)
- Condition information to segments
    - Remove idle time (acceleration ≅ gravity for continuous time)
    - Add compass heading (field strength ≅ region's magnetic field)
        - 30-50 μT for North-East USA

After Drift Reduction



After Alignment



Search Algorithm

- Goals and theorems
  - Find sequence of turns (θ) in graph (G) that maximize probability of matching observed turns (α)
  - If turn errors approximate to a zero-mean Gaussian distribution (mean = 0 and std dev = σ)
    - Maximizing the probability of optimal route is equivalent to minimizing the L2 norm of the error ($||\alpha - \theta||$)
    - The optimal route tracking solution becomes $max(||\alpha - \theta||)$ for all $\theta \in G$
- Based on ‘Trellis Code Decoding’ technique
  - More complex as start segment not known
  - Improved results by filtering unlikely connections
- Individual and Cluster Rank metrics
  - Identify individual routes traversed
  - Cluster similar routes to increase confidence in an area

Search Algorithm (contd.)

- The algorithm
  - Assume each segment as a potential starting point
  - Iterate through each potential path (for every intersection)
    - Filter out all unlikely connections
    - Score remaining connections (add previous score)
  - Pick top scoring paths (trade-off between speed and accuracy)

- Filtering out unlikely connections
  - Reported turn angle - Connection turn angle < Turn threshold
  - Reported segment heading - Connection heading < Heading threshold *(if stable)*
  - Reported travel time < Minimum time between intersections

Scoring

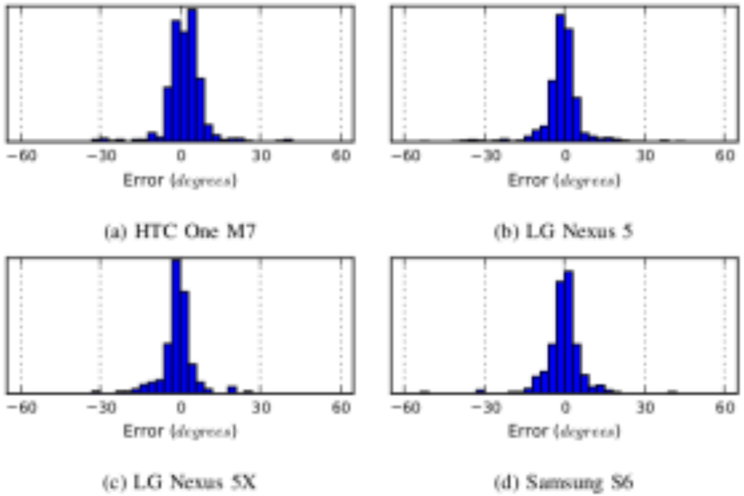
- Based on weighted turn angles, curvature and travel time
  - Turn Score = Turn weight * abs(Reported turn angle - Connection turn angle)
  - Time Score = Time weight * abs(Reported travel time - Minimum time between intersections)
- Curvature Scoring
  - Split graph segment curvature into equal parts as Gyroscope segment curvature
    - Assume constant velocity
  - Calculate normalized distance between segment and Gyroscope curve for each part
    - Curve Score = (1 / Segment time) * sum(abs(Reported curve - Segment curve) for all parts)
- L2 norm theoretically optimal for Gaussian distributions, however
  - L1 norm preferred over L2 norm (Gyroscope errors not truly Gaussian)
  - L2 squaring amplifies sparse large errors

Final score = Sum of (Turn + Time + Curve) score for all intersections

Evaluation Metric - Gyroscope Accuracy

- **Error distribution** used to check accuracy
  - From real driving experiments
  - Error = (Reported turn angle - OSM turn angle)
- Key Results:
  - Distributions resemble Gaussian distribution
  - **~ 95% of errors less than 10°**

Error Distribution for four smartphones

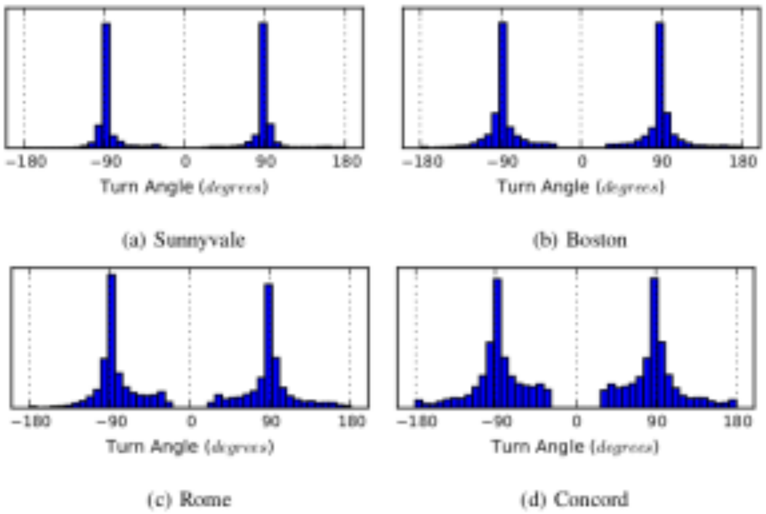


Phone	No. Turns N	Mean μ	Std. dev. σ
HTC One M7	482	1.73°	7.07°
LG Nexus 5	618	-0.77°	7.89°
LG Nexus 5X	170	-1.12°	6.40°
Samsung S6	238	-0.57°	7.51°

Cities for Simulation

- 11 cities for simulation
  - Based on size, density and road structure
- Large number of Vertices V and Edges E
  - Signifies big cities with low inference potential
- Disparate turn distribution
  - Signifies unique turns with high inference potential
- Many similar turn radii
  - Signifies grid-like with low inference potential

Turn Distribution for four cities



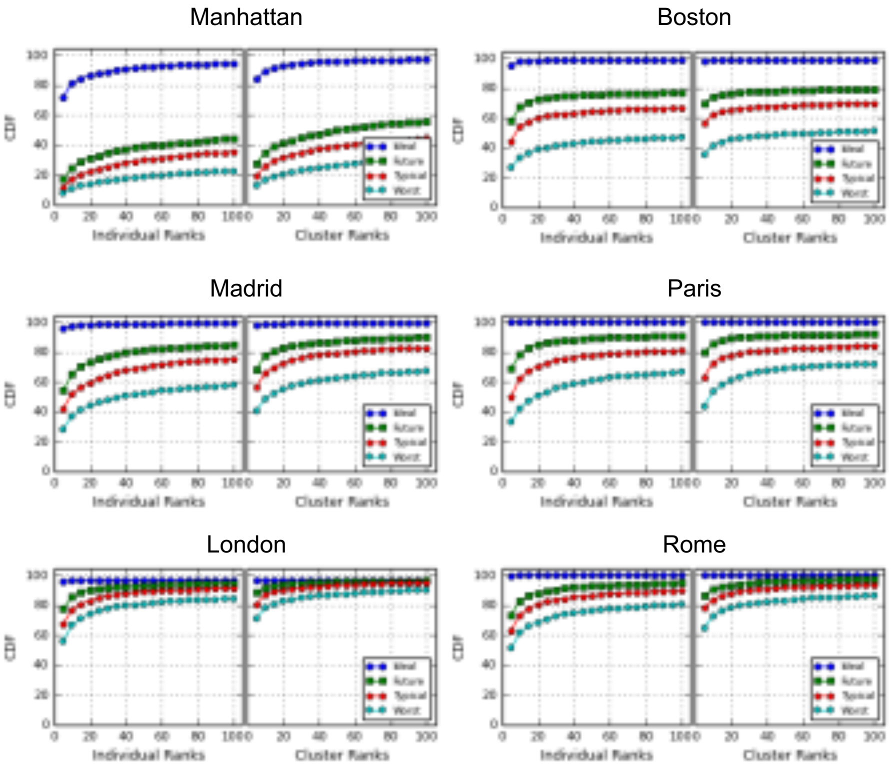
City	$\|V\|$	$\|E\|$	Mean μ_{turn}	Std Dev σ_{turn}
Atlanta, GA, USA	10529	25557	88.73°	17.58°
Berlin, Germany	4708	19752	88.21°	19.87°
Boston, MA, USA	8010	22149	89.69°	20.52°
Concord, MA, USA	3049	6467	88.13°	29.58°
London, UK	9468	21968	87.83°	20.38°
Madrid, Spain	10012	30144	86.41°	25.13°
Manhattan, NY, USA	1033	3699	89.23°	17.81°
Paris, France	6744	11204	86.35°	26.26°
Rome, Italy	9408	20577	85.98°	26.15°
Sunnyvale, CA, USA	5592	12302	88.59°	16.00°
Waltham, MA, USA	3366	9437	88.93°	20.53°

Creating Simulation Routes

- Creating simulation routes
  - Connect segments starting at a random start segment
  - Inject variable noise (turn, curve & time) to simulate real driving routes
- Noise scenarios
  - <u>Ideal</u> (noise free scenario)
  - <u>Typical</u> (moderate traffic and current sensors)
    - Using values from real driving experiments
  - <u>High Noise</u> (heavy traffic and less accurate sensors)
  - <u>Future</u> (moderate traffic and more accurate sensors)

Evaluation Metric - Simulation Routes

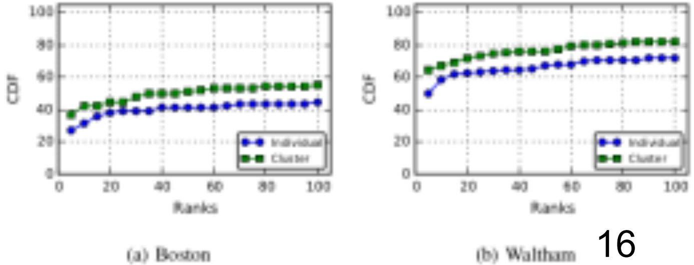
- 8000 routes for each city
  - 2000 routes * 4 noise scenarios
- Key results
  - Good inference for 8 cities (Individual / Cluster)
    - <u>Typical scenario:</u> 50 / 60% in top 10
    - <u>High noise scenario:</u> 35 / 40% in top 10
  - Low inference for grid-like cities
    - E.g. Manhattan
  - Turn & curvature combined have largest impact
    - E.g. London and Rome
    - Boston, Madrid and Paris have straight roads
  - Size of city doesn’t impact inference



Evaluation Metric - Real Driving Routes

- 70 routes each in Boston & Waltham (~ 980 km)
  - Restrictions - Fixed Position and no reversal
- Key results
  - Boston
    - ~ 30 / 35% in top 5 (13% ranked 1)
    - Leans toward high noise scenario of simulation
  - Waltham
    - ~ 50 / 60% in top 5 (38% ranked 1)
    - Leans toward typical noise scenario of simulation

Real Driving Experiments Results



Summary

- Demonstrated that apps with no permissions can infer routes with good accuracy
- Used graph theory to identify the most likely routes and clusters
- Collected 140 driving experiments (~980 km) for Boston and Waltham
- ~ 30% of routes in top 5 for Boston and 50% in top 5 for Waltham
- Performed simulations for 11 cities with diverse road characteristics
- Good inference for 8 cities in simulation with more than 50% of routes in top 10

Thank You

Questions?