CrowdSensing

Qian Zhang
Agenda

01 Overview for crowdsensing
02 Crowdsensing applications
03 Urban sensing
Introduction

• 20-30 billions of devices by 2020
• Smartphones comes with various sensors: Camera, accelerometer, GPS, compass, etc.
• Mobile devices: more powerful (more computing, communication and storage capabilities)
• Mobile Crowdsensing means the integration of sensors that can be used for gathering materialistic or non-materialistic information
• Needs people who use these sensors & obviously their global participation
Phenomena

Individual
- movement patterns, modes of transportation, and activities.

Community
- pollution (air/noise) levels in a neighborhood, real-time traffic patterns, pot holes on roads, road closures and transit timings.

Introduction
Introduction

• Community sensing is popularly called *participatory* sensing or *opportunistic* sensing
  - **Participatory sensing** - individuals are actively involved in contributing sensor data
  - **Opportunistic sensing** - autonomous and user involvement is minimal

• Research Challenges
  - Localized analytics
  - Resource limitations
  - Privacy
  - Aggregate analytics
  - Architecture
From Participatory Sensing to Mobile Crowd Sensing

- Both crowd wisdom and crowdsourcing rely on human intelligence.
- Participatory sensing and MCS explore a fusion of human and machine intelligence.
A Reference Framework for Mobile Crowd Sensing

data visualization and user interface

learning and inference techniques

gathers data from selected sensor nodes and provides privacy-preserving mechanisms for data contributors

users can decide to whom her data can be shared
MOBILE CROWDSENSING APPLICATIONS

- Environmental
  - Natural Environment
- Infrastructure
  - Public Infrastructure
- Social
  - Personal information
MCS: UNIQUE CHARACTERISTICS

• Multi-modality sensing capabilities
• Deployed in the field
• The dynamic conditions in the collection of mobile devices
• Privacy
• Energy
• Cost
• Efforts
Typical Functioning of Mobile Crowdsensing

Raw sensor data are collected on devices and processed by local analytic algorithms to produce consumable data for applications. The data may then be modified to preserve privacy and is sent to the backend for aggregation and mining.
Types of crowdsensing

On the basis of the type of the measured phenomenon

1. Environmental crowdsensing: used for measuring the natural environment (e.g., level of water, air pollution, wildfire habitats).
2. Infrastructure crowdsensing: used for measuring the public infrastructure (e.g., traffic congestion and road conditions).
3. Social crowdsensing: used for measuring data about the social life of individuals (e.g., the cinemas visited by an individual).
Environmental Monitoring Using Crowdsensing

1. Monitoring Air Pollution Level
2. Monitoring Noise Pollution
Monitoring Air Pollution Level

- Air pollution causes various types of respiratory diseases, cancer and also causes acid rainfall
- It is important to have a good pollution map available to the public
- Large ultrafine particles which are spread widely are responsible for negative effects in the human health
- Mobile measurement system can effectively be used to derive accurate ultrafine particles pollution maps with high spatio-temporal resolution
- Sensors were installed on the public vehicles in order to measure more than a year in Zurich
- Database are locally stored and transmitted to server running Global sensor network through cellular networks. The data is cleared from the local database once acknowledged about the receiving of data by the server
Findings:

• The particle concentrations are higher during the week (Monday to Saturday) than on Sunday due to higher traffic volumes.
• Pollution levels are higher in winter and fall than in spring and summer.
• Different factors like terrain elevation, building heights and traffic density have larger impact on the predicted pollution level.
Monitoring Noise Pollution

• Participatory crowdsensing can help in monitoring noise pollution in low cost
• Application called NoiseTube uses smartphones as the instrument for measurement of noise pollution
• Users provides the information like source, location and time of noise
• Components of the NoiseTube platform are a mobile application which can be downloaded and installed freely on smart phones and a Web-based community memory system running on a central server
• Application interface includes the following main components: the measured sound level (visualized with a histogram and a color scale), complemented by a dose indicator (on top), the tagging component and the location tagging component (for indoor locations for instance)
Monitoring Noise Pollution

- The GPS receiver is used by the application for every measurement to include the coordinates of the geographical location.
- The social tagging makes users to tag the level of sound that is measured through the application.
- Web based community memory gathers all the data of noise pollution and also provides the tools for exploring, visualizing, analyzing and searching of data.
- Right figure shows the NoiseTube interface

![NoiseTube Interface](image)
Challenges of Crowdsensing

• User participation:
  • Performance and usefulness of such sensor networks heavily depends on the crowd sensor’s willingness to participate in the data collection process

• Data sensing quality:
  • The obvious question is how to validate the sensing data that crowd sensors provide to the system

• User anonymity:
  • GPS sensor readings can be use to track users movements and profile them for other purposes besides their crowd sensing tasks
Agenda

01 Overview for crowdsensing

02 Crowdsensing applications

03 Urban sensing
Crowd-Sensing for On-street Smart Parking

(Shawn) Xiao Chen, Elizeu Santos-Neto, Matei Ripeanu

Electrical and Computer Engineering Department
University of British Columbia
Overview

What is smart parking and its objectives?
What are the current solutions and their problems?
What is our proposed solution and its advantages?

How can the organizer guide the data collection?
How should participants respond to contribute data?
How should we deal with free riders?

Why should we prefer coordinated crowdsourcing?
Why can we simplify users’ manual operation?
Why we should not always exclude free riders?
Parking problem / Smart parking

- Searching for free parking spots costs billions:
  - congested traffic (30%)
  - pollution,
  - wasted time and fuel

- Smart Parking:
  - collect real-time data on parking availability,
  - guide drivers to find free spots efficiently.
Objectives

Compared with ordinary drivers

- Cruising Time
- Walking Distance

Compared with ordinary drivers
Data collection: Infrastructure-based approaches

- Infrastructure to detect status of parking slots (sensor or RSU)
  - Collect and distribute data

- High initial investment and maintenance cost
  - Suitable for indoor garage or large parking lots
  - $20/month/spot

- Example: SFParking
  - Deployed in San Francisco
Data-collection: CrowdSensing approaches

• Collect relevant data from the public through their mobile phones
  – (almost) no initial investment,
  – but dependent on users’ manual input

• Example: Google’s Open Spot
Problems with current approach

• Difficult to use
  – apart from navigation, too much info to read/enter

• Limited info
  – only from previous contributors, no info about occupied streets

• Uncoordinated
  – race for the same spot, users not willing/guided to explore unknown areas
System Components

- Central Server
  - Search result
  - Update annotation
  - Where to park
- Smart Parker
  - Manual input before driving or after parking
- Client Device
  - Driving instructions for parking
  - Automatic sensing
- Sensor data & Driver answers
Potential Advantages

- Easy to use
  - Integrated with road navigation system
- Guided parking
  - By coordinating drivers
- Higher adoption
  - Mutual assistance, resilient to free riders
Design alternatives

**What**
- What is smart parking and its objectives?
- What are the current solutions and their problems?
- What is our proposed solution and its advantages?

**How**
- How can the organizer guide the crowd-sourced data collection?
- How should participants respond to contribute data?
- How should we deal with free riders?

**Why**
- Why should we prefer coordinated crowdsourcing?
- Why can we simplify users’ manual operation?
- Why we should not always exclude free riders?
Data Acquisition

• Types of questions to ask smart parkers

<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
<th>Answers</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>How many parking spots on street?</td>
<td>0,1,2,3...</td>
<td>As the answer</td>
</tr>
<tr>
<td>Q2</td>
<td>Any parking spots on the street?</td>
<td>Yes/No</td>
<td>1(Yes)/0(No)</td>
</tr>
<tr>
<td>Q3</td>
<td>No question / inference</td>
<td>No answer</td>
<td>Always 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Observed behavior</th>
<th>Inference</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Reach the assigned street and continue at low speed</td>
<td>The assigned street is occupied</td>
<td>0</td>
</tr>
<tr>
<td>I2</td>
<td>Move at low speed after I1</td>
<td>The past street is occupied</td>
<td>0</td>
</tr>
<tr>
<td>I3</td>
<td>Launch the application and drive away</td>
<td>New vacancy in the street</td>
<td>+1</td>
</tr>
</tbody>
</table>
What
What is smart parking and its objectives?
What are the current solutions and their problems?
What is our proposed solution and its advantages?

How
How can the organizer steer the crowd-sourced data collection?
How should participants respond to contribute data?
How should we deal with free riders?

Why
Why should we prefer coordinated crowdsourcing?
Why can we simplify users’ manual operation?
Why we should not always exclude free riders?
Coordination is necessary

- When uncoordinated, smart parkers fail to find parking slots closer to their destination than ordinary drivers.
Coordinated smart parking works!

- When coordinated, a majority of smart parkers don’t need to cruise for the parking slots.
- Even those who need to cruise spend far less time than ordinary drivers.
Manual operation can be simplified

- With high adoption the service is functional with only answering simple questions
  - When the percentage of smart parkers is low, inference by sensor data becomes useful
Accept freeriders!

- As the number of free-riders grows, the quality of service deteriorates only slowly.
- When there are sufficient contributors, social benefits grow as more people free-ride.
Summary

• Coordination is key to effective parking guidance
• CrowdSensing: Simplified input is enough if there are enough participants
• Accepting free riders increases social benefits (if there are some contributors)
Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Tao Wang, Xiaoming Li
EECS School, Peking University, China
ECE Dept., Stony Brook University

ACM MobiCom 2014
Jigsaw: Floor plan reconstruction

Motivation
Jigsaw: Floor plan reconstruction

Motivation

Indoor Maps availability
Jigsaw: Floor plan reconstruction

Motivation

• Crowdsensing based construction
  • Gather piecewise data from individual mobile users
    • e.g., images, inertial sensor data
  • Extract floor plan information
  • Put pieces together into a complete floor plan
• Benefits
  • Service providers (e.g., Google) don’t need to negotiate with building owners one by one
  • No need to hire dedicated personnel for inch-by-inch measurements either
Crowsensing to construct floor plan

• Challenges
  • Accurate coordinates and orientations of indoor landmarks (i.e., POIs such as store entrances)
    • Inertial data couldn’t provide
  • Insufficient “anchor points”
    • Error accumulation in dead reckoning
    • Over- and under- estimation of accessible areas

• Inspiration
  • Complementary strengths of vision and mobile techniques
    • Vision ones to produce accurate geometric information for landmarks
    • Inertial data to obtain placement of landmarks, and less critical hallway and room shapes
  • Use optimization and probabilistic formulations
    • Robustness against errors/noises from data
Jigsaw overview

• Three stages
  • Landmark modeling: extract landmark geometry from images
  • Landmark placement: obtain pairwise landmark spatial relation (e.g., distance, orientation) from inertial data
  • Map augmentation: construct hallway and room shapes from mobile traces
Landmark modeling

- **Goal**
  - Extract sizes and coordinates of major geometry features (e.g., widths of entrances, lengths/orientations of walls) of landmarks
- **Method:** extend two computer vision techniques
  - **Structure from Motion (SfM):** given a set of images of the same object from different viewpoints, generate (in the LOCAL coordinate system)
    - 1) a “cloud” of 3d points representing the exterior shape of the object;
    - 2) the location where each image is taken
  - **Vanishing line detection:** given an image, detect orthogonal line segments of the object
Landmark modeling process (1/2)

- Geometric vertices
  - **P**: four corners of a store entrance
  - **Q**: connecting points of wall segments

- Extract the coordinates of geometric vertices
  - Step 1. Extract landmark’s major contour lines on each image

- Step 2. Project 2D lines into 3D
  - Project 2D lines using transformation matrices by SfM
  - Use adapted k-means to cluster major geometry lines
Landmark modeling process (2/2)

- Detect connecting points of wall segments
  - Project the 3d point cloud onto XY plane
  - Detect wall segments and their connecting points
    - Use entrance line (P3P4) from the previous step as the start
    - Find the two ends (Q1Q2)
    - Continue to search for more connecting point (Q3)
Landmark placement

• Goal
  • Input: landmark models in their local coordinate systems
    • Major geometry features, positions of cameras
  • Output: landmarks placed on a global coordinate system
    • Absolute coordinates and orientations

• Method
  • Step 1. Obtain pairwise spatial relationship between adjacent landmarks
  • Step 2. place adjacent landmarks on the common ground
Micro-tasks for spatial relationships

- A series of data gathering actions
  - Obtain pairwise distance and orientation constraints
- Click-Rotate-Click (CRC)
  - $\omega$: rotated angles from gyroscope
  - $\langle d_A, \beta_A \rangle$ and $\langle d_B, \beta_B \rangle$: SfM output
  - Relative distance and orientation between A,B uniquely determined
- Click-Walk-Click (CWC)
  - $|C_AC_B|$: step counting
  - $\omega_A$ and $\omega_B$: placement offset estimation and gyroscope readings
  - $\langle d_A, \beta_A \rangle$ and $\langle d_B, \beta_B \rangle$: SfM output
  - Similar measurements calculation
Micro-tasks for spatial relationships

- A series of data gathering actions
  - Obtain pairwise distance and orientation constraints
- Click-Rotate-Click (CRC)
  - $\omega$: rotated angles from gyroscope
  - $(d_A, \beta_A)$ and $(d_B, \beta_B)$: SfM output
  - Relative distance and orientation between A, B uniquely determined
- Click-Walk-Click (CWC)
  - $|C_A C_B|$: step counting
  - $\omega_A$ and $\omega_B$: placement offset estimation and gyroscope readings
  - $(d_A, \beta_A)$ and $(d_B, \beta_B)$: SfM output
  - Similar measurements calculation
Landmark placement formulation

• Multiple distance and orientation constraints

• Maximum Likelihood Estimation (MLE)
  • Θ*: the most likely coordinates and orientations
    • Θ = {X, ϕ}: coordinates and orientations of landmarks
    • Z, O: observations of X, ϕ

\[ \theta^* = \arg \max_{\theta} P(Z, O | X, \phi) \]

• Landmark placement results
Hallway boundary construction

- Two connection options
  - Direct line between two segments
    - collinear or facing each other
  - Extend two segments to an intersection point
    - Perpendicular walls
Hallway boundary construction

• Two connection options
  • Direct line between two segments
    • collinear or facing each other
  • Extend two segments to an intersection point
    • Perpendicular walls

• Problem formulation
  • Minimum weight matching in a bipartite graph

• Solution: Kuhn-Munkres algorithm*
  • $O(n^3)$, $n$: number of landmarks

Compare with alternative methods

• Naïve convex hull
  • Miss segments inside

• Greedy algorithms
  • Depend on order of connecting
  • Miss 90° corners

• Our results
Details reconstruction: hallway shape

• Step 1. build *occupancy grid map*
  • Grid cells each with a variable representing the probability it is accessible
  • a) External boundary of hallway
  • b) Camera positions
  • c) Trajectories
Details reconstruction: hallway shape

- **Step 1. build occupancy grid map**
  - Grid cells each with a variable representing the probability it is accessible
  - a) External boundary of hallway
  - b) Camera positions
  - c) Trajectories

- **Step 2. Binaryzation with a threshold**

- **Step 3. Smoothing**
  - Alpha-shape*

Details reconstruction: room shape

- Room reconstruction
  - Data-gathering micro-task
    - CWC inside one room
  - Step 1. determine initial/final locations
    - Two camera locations as anchor points
Details reconstruction: room shape

- Room reconstruction
  - Data-gathering micro-task
    - CWC inside one room
  - Step 1. determine initial/final locations
    - Two camera locations as anchor points
  - Step 2. use trajectories to build an occupancy grid map
  - Step 3. similar thresholding and smoothing
- Results

![Room reconstruction images]

- Stores
- Combined hallway, stores
Evaluation

• Methodology
  • 3 stories of malls: 150x75m and 140x40m
  • 8,13,14 store entrances as landmarks
  • 150 photos for each landmark
  • 182,184,151 CRC measurements
  • 24 CWC measurements in story 3
    • Comprised of two parts
  • 96,106,73 user traces along hallway
  • ~7 traces inside each store

• Floor plans
Reconstructed floor plans

- Landmark placement performance
  - Store position error 1-2m
  - Store orientation error 5-9 degrees
Reconstructed floor plans

- Landmark placement performance
  - Store position error 1-2m
  - Store orientation error 5-9 degrees
- Constructed floor plans
Detailed results

- **Accuracy of floor plans**
  - Root mean square error (RMSE)
    - $X_i=(x_i,y_i)$: 2D coordinates
  - Features
    - Landmarks
    - Hallway intersections

- **Hallway shape**
  - Overlay the reconstructed hallway onto its groundtruth to achieve maximum overlap
  - Hallway shape
    - Precision~80%, Recall~90%, F-score~84%
Comparison with CrowdInside++

• Several assumptions of CrowdInside*
  • Sufficient numbers of anchor points (GPS, inertial, ..)
  • Sufficient amount of traces passing through anchor points
  • Distinctive WiFi signatures in different rooms

• Artificial improvements in CrowdInside++
  • Double the number of anchor points; assume they are GPS-based
  • All traces pass through adjacent anchor points
  • Manually classify room traces

• Results of CrowdInside++
  • Miss a few small-sized stores
  • RMSE and maximum error: 4x of Jigsaw
  • Hallway shape: ~30% less than Jigsaw

Comparison with CrowdInside++

• Several assumptions of CrowdInside*
  • Sufficient numbers of anchor points (GPS, inertial, ..)
  • Sufficient amount of traces passing through anchor points
  • Distinctive WiFi signatures in different rooms

• Artificial improvements in CrowdInside++
  • Double the number of anchor points; assume they are GPS-based
  • All traces pass through adjacent anchor points
  • Manually classify room traces

• Results of CrowdInside++

• Causes
  • Error accumulation of inertial-only approach
  • Deterministic alpha-shape instead of probabilistic occupancy map

Related work

• Floor plan construction: relatively new problem
  • CrowdInside, Jiang et. al., Walkie-Markie, MapGenie
    • We combine vision and mobile techniques
    • We use optimization and probabilistic techniques
• SLAM
  • Noisy and piece-wise crowdsensed data
    • No high precision special sensor: laser ranges, stereo/depth cameras
    • Estimate landmark orientations
• 3D construction in vision
  • Floor plans require only 2d
• Localization with vision techniques
  • Sextant, OPS
Summary

• Combine complementary strengths of vision and mobile techniques
  • Vision: accurate geometric information, landmark only
  • Mobile: relative positions of landmarks, sketches of hallway/room shapes
  • Camera locations as anchor points
• Optimization and probabilistic formulations for solid foundations and better robustness
  • MLE: landmark placement
  • Minimum weight matching: hallway boundary construction
  • Occupancy grid map: hallway/room shapes
Enabling Physical Analytics in Retail Stores

Online sales in U.S. in 2013 only 11%

Using Smart Glasses

Important to capture shopper behavior not only in the online world but also in the physical world

Physical Analytics

Understanding the intent of the shoppers in the physical world

Benefit

Enable contextual recommendations

Shopping list reminders

Guides to product locations

Alice sees a coupon as she is about to walk away!
Contributions

• Fuse Wi-Fi, inertial sensor and video data from smart glasses
• AutoLayout: Map the store without any user or store input
• Use these inferences to track glass/non-glass users in online phase
• Characterize walk, dwell, gaze and reaching-out activities of shoppers
• Attention identification within the captured frame
**Technology**

Localization, Product layouts, User analytics

**Incentives**

- **Stores**: increased sales
- **Physical analytics provider**: share of profits by partnering with stores
- **Users**: discounts, shopping

**Privacy**

Shoppers Willing to Tell All

Teresa Novellino
Upstart Business Journal Entrepreneurs & Enterprises Editor
Email | Twitter

It might surprise retailers, but a new IBM study reveals that consumers are much more willing to give up information about themselves.
Overview: ThirdEye - AutoLayout

Few Glass users
Glass "sees" products

Many Non-Glass users

Key Idea:
- Wi-Fi
- Inertial sensor
- Video

Offline: AutoLayout
Localize Glass users
Build product map
AP locations

Online

Localize Non-Glass users
Offline Phase (AutoLayout)
Problem formulation: unknowns

\( sLoc^i_k \): 2D location of \( k^{th} \) shopper after \( i^{th} \) step

\( pLoc_j \): 2D location of \( j^{th} \) product in store

\(< P_l, aLoc_l, \gamma_l >\): parameters of \( l^{th} \) access point

Path loss constant
Access point location
Transmit power

Critical to localize non-glass users!
Problem formulation

Minimize

\[ w_1 \cdot r(s\text{Loc}, a\text{Loc}, P, \gamma) + w_2 \cdot p(s\text{Loc}, t) + w_3 \cdot q(s\text{Loc}, p\text{Loc}) \]

Log Distance Path Loss (LDPL) model:

\[ eRSS(s\text{Loc}, a\text{Loc}, apTxPwr, \gamma) = apTxPwr - 10 \gamma \log(|| s\text{Loc} - ap\text{Loc} ||) \]

Minimizes error in measured RSS values and those estimated by parameters describing the LDPL model

\[ \sum_l \sum_{l,k} \sum_{i} \left\| mRSS_{l,k,i} - eRSS(s\text{Loc}_k^i, a\text{Loc}_l, P_l, \gamma_i) \right\| \]

over all measured RSS values from that AP across all users.
Incorporate mobility: inertial sensors

Minimize

- Wi-Fi term
- Inertial sensor term
- Camera term

\[ w_1 \cdot r(s\text{Loc}, a\text{Loc}, P, \gamma) + w_2 \cdot p(s\text{Loc}, t) + w_3 \cdot q(s\text{Loc}, p\text{Loc}) \]

Accelerometer: step-count \([\text{Zee}, \text{UnLoc}] \rightarrow \text{distance}\)
Compass: heading direction

For all shoppers, at all steps:
- \( x_{i+1} \approx x_i + d \cdot \cos(\theta) \)
- \( y_{i+1} \approx y_i + d \cdot \sin(\theta) \)

\[
\sum_i \sum_k \| s\text{Loc}^{i+1}_k - s\text{Loc}^i_k - \hat{e}^i_k \|^2
\]

\( i^{th} \) step \( k^{th} \) shopper

\( \hat{e}^i_k = [\cos \theta^i_k \sin \theta^i_k]^T \)
Tie in product locations: camera

Minimize

\[ w_1 \cdot r(s\text{Loc}, a\text{Loc}, P, \gamma) + w_2 \cdot p(s\text{Loc}, t) + w_3 \cdot q(s\text{Loc}, p\text{Loc}) \]

Leverage Google Reverse Image search to obtain labels for product images

All shopper locations from where a particular product was seen must be close to each other

\[ \sum_j \sum_{<k,m> \in L_j} \|s\text{Loc}_k^m - p\text{Loc}_j\|^2 \]

\(j^{th}\) product \(k^{th}\) shopper saw \(j^{th}\) product at \(m^{th}\) step
Optimization

Leverage product locations to align different walks

- Origin: $sLoc_0^0 = (0,0)$
- Leverage mobility: Locations within a walk are connected via inertial sensor data

Milk seen by shopper 0 and shopper 2

- Run BFS to initialize all shopper & product locations
- Gradient descent: refine initial estimates
Example walks around aisles in Target

Actual walks around aisles

After BFS: all tracks are in same coordinate system

After optimization: tracks look closer to actual walk
Inferred Layout for H-E-B

Improves with more shoppers
AutoLayout

Behavior Classification
Overview: ThirdEye - User Analytics

In a retail setting

Walk

Dwell

Gaze

Reach-out
Behavior Classification Algorithm

Start

Shopper’s head steady? (Inertial sensing)

Yes

Shopper’s view is Steady (Video)

Yes

Is there a hand near the items (Video)

Yes

Reach Out

No

Gaze

No

Dwell

No

Walk

Shopper’s velocity < threshold (Inertial sensing)
Gaze and Reach-out

When shopper is gazing/reaching-out scene in front of him does not change

Leverage vision based technique Optical Flow to detect gaze/reach-out

- Optical flow (of): difference in terms of pixels between consecutive images
- If \( of < of_{gaze} \) detect gaze/reach-out

88% detection rate at 1.8% false detections
Reach-out detection

Reach-out indicates high degree of interest: important to detect

Hand seen in the frame:
— detect hands to detect reach-out

Train TextonBoost Classifier
Leveraging TextonBoost classifier

Divided hand

Cluster together nearby segments

Spurious hand

Ignore very small segments

Detection success rate: 86% False detection rate: 15%
Dwell detection

Accelerometer showing that user is static?

Shopper may not be static, he may take few steps looking at nearby items

Dwell characterized by small net displacement!

Detect dwell based on periods of low net displacement

Suppose K steps in time window $\tau$ and heading at step $i$ is $\theta_i$
— Detect steps using prior work Zee [MobiCom 2011]

Magnitude of net velocity vector

$$||v|| = \sqrt{\left(\sum_{i=1}^{K} \cos \theta_i \right)^2 + \left(\sum_{i=1}^{K} \sin \theta_i \right)^2}$$

95% detection rate at 10% false alarms

Dwell if:

$$||v|| < ||v||_{dwell}$$

Net displacement in 5 sec
Conclusion

Our contributions

— Fuse Wi-Fi, inertial sensor and video data from smart glasses
— **AutoLayout**: Map the store without any user or store input
— Use these inferences to track glass/non-glass users in online phase
— Characterize walk, dwell, gaze and reaching-out activities of shoppers
— Attention identification within the captured frame

Future work

— Larger data-set for patterns representative of more diverse population
— In-depth analytics of shoppers
Agenda

01  Overview for crowdsensing
02  Crowdsensing applications
03  Urban sensing
Urban Sensing Based on Human Mobility

Shenggong Ji, Yu Zheng, Tianrui Li

• Southwest Jiaotong University, Chengdu, Sichuan, China
  • Microsoft Research Asia, Beijing, China
Urban Sensing

• Collecting urban data
  – Noise, temperature, air quality, …
  – Human as a sensor

• Brings challenges to
  • City-scale real-time monitoring
  • Further data analytics

Skewed human mobility

Imbalanced data coverage
An Urban Sensing Framework

- Consider real-world human mobility
- Maximize the amount and balance of collected data
- Given a limited budget

Participant Recruitment and Task Design

Recruiting $u_1$ and $u_2$ with tasks

Unit reward for each hour

Collected Data

Time Span: 7-10am
Challenges

- Measure data balance: different spatio-temporal granularities

- High computational cost
  - Task design for a participant (routing planning)
  - Recruiting participants from many candidates
Framework

- A participant recruitment mechanism
  - random recruitment
  - replacement-based refinement

- A task design algorithm
  - A hierarchical entropy-based objective function
Hierarchical Entropy-based Objective Function

\[
\max \phi = \alpha \times E + (1 - \alpha) \times \log_2 Q
\]

\(\alpha\): the relative preference of data balance to data amount
- application specific

Coarse-grained partition

Fine-grained partition
Designed Task: (9:00, 3) → (9:04, 6) → (9:08, 7)
Evaluation

• Datasets
  – Human mobility dataset from a real-world noise sensing experiment
    • Sensing region: 6.6km × 3.3km
    • Sensing time interval: 6:00 am ~ 22:00 pm
  – 244 participant candidates with mobility information

• Settings
  – Hierarchical partitions for data coverage
    • \( I(k) \times J(k) \): spatial partition
    • \( T(k) \): temporal partition

<table>
<thead>
<tr>
<th>Granularity ( k )</th>
<th>( I(k) )</th>
<th>( J(k) )</th>
<th>( T(k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
Evaluation

• Collecting data with a good coverage
  • Even with skewed human mobility
  • $\phi = \alpha \times E + (1 - \alpha) \times \log_2 Q$

• Result:
  • $\alpha = 0$: most amount
  • $\alpha = 1$: most balancing
Evaluation

• Participant recruitment mechanism
  • Ours: Random recruitment + Replacement-based refinement
  • Two baselines for comparison
    1. Random recruitment
    2. Greedy recruitment

<table>
<thead>
<tr>
<th></th>
<th>$E(A(1))$</th>
<th>$E(A(2))$</th>
<th>$E(A(3))$</th>
<th>$Q(A)$</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>9.84</td>
<td>8.35</td>
<td>5.93</td>
<td>1780</td>
<td>≈ 5 minutes</td>
</tr>
<tr>
<td>Greedy</td>
<td>9.94</td>
<td>8.46</td>
<td>6.04</td>
<td>1847</td>
<td>≈ 55 minutes</td>
</tr>
<tr>
<td>Ours</td>
<td>10.08</td>
<td>8.56</td>
<td>6.05</td>
<td>2053</td>
<td>≈ 8 minutes</td>
</tr>
</tbody>
</table>

• Results
  • Data coverage: best performance
  • Running time: very efficient
Conclusion

• We proposed a novel urban sensing framework

• Methodology
  • A participant recruitment mechanism
  • A hierarchical entropy-based objective function
  • A graph-based task design algorithm

• Extensive experiments using real-world human mobility

• Collecting data with better (more balanced) coverage

• Data Released:
End of This Chapter