AutoWitness: Locating and Tracking Stolen Property While Tolerating GPS and Radio Outages

Santanu Guha*, Kurt Plarre*, Daniel Lissner*, Somnath Mitra*, Bhagavathy Krishna*, Prabal Dutta~, Santosh Kumar*

*University of Memphis    ~University of Michigan
In United States:
Over 2 million reported burglaries in 2009
~ An average $2000 loss per incident
Existing Theft Detection Systems

- Either deter or detect burglary incidents
Existing Tracking Systems

- Radio Outages
- Unsuitable for smaller assets
We need a system that is...

- Immune to Radio Outages
- Stealthy
- Long Life

- 1 year
- 2 years

- Long Life
Our “AutoWitness” system

Detects Burglary Autonomously

• Using Vehicular Movement as an identifier of theft

Tracks Asset, Pinpoints Final Location

• Using a HMM based model to track burglars using only inertial estimates.
AutoWitness

Tag Node
- Embedded in expensive items

Server
- Used for Map Matching
Tag Node inside the Stolen Item in Burglar’s Car

Distance and Turn Estimates

AutoWitness Server

Position and Travelled Path of Burglar’s Car

Cell Tower

Burglar’s Car

Police Car
How does AutoWitness work?

Autonomous Detection of Burglary by the Tag Node
Vehicular Movement Indicates Theft
How does AutoWitness work?

Autonomous Detection of Burglary by the Tag Node

Detection of Theft initiates tracking of assets
How does AutoWitness work?

Autonomous Detection of Burglary by the Tag Node

Detection of Theft initiates tracking of assets

Track is provided to law enforcement officials when cell tower is available completing the process of asset recovery
Cell Tower

AutoWitness Server

Map Matching
  • HMM
  • City Map

Asset Location

Distance: Accel
Turns: Gyro
Thank you AutoWitness!

- Classifies Theft by Detecting Vehicular Signature
- Produce inertial estimates using accelerometers and gyroscopes
- Computes track of burglar using a HMM
- Informs the Police
Key System Challenges: AutoWitness

Real Life Deployment presents several system challenges

Challenges
• Tag Node
• Server
Key System Challenges: Tag Node

- Producing Accurate Inertial Estimates
- Choosing appropriate Hardware
- Re-orientating Tag Node
- Developing Theft Classifier
Choice of Hardware for Tag Node

Wake Up Circuit

Vibration Dosimeter
- Filters out insignificant vibrations and prolongs tag node lifetime
Choice of Hardware for Tag Node

Distance = d

Angle = Θ

- 3 axis accelerometer
- 3 axis Gyroscope
Choice of Hardware for Tag Node

- A GSM / GPRS modem
All integrated in a Epic Core Platform
Light Weight detection of burglary on Tag Node

- Significant Vibrations
- Vehicular movement is taken as indicator of theft
- Extensive Data Collected for different movement scenarios
- Classifier Built using 10 fold cross validation in Weka
Producing Accurate Inertial Estimates

- Distance Computation using inertial sensors
  - Remove noise
  - Re-orient Tag Node
  - Correct for Radial Acceleration
  - Correct for drift
Distance Computation

- Remove Noise
- Subtract Mean between stops to correct drift
- Double Integrate

Also remove component radial acceleration for curves
Distance Computation

Distance Error < 12.6 %
Key System Challenges: Server Side

Using only Inertial Estimates

Pinpoint the asset’s final location

Reconstruct the path driven by burglar
Reconstruct Path driven by Burglar

Hidden Markov Model

- $P(S_i|\text{Start})$
- $P(O_i|\text{State } i)$
- $P(\text{State } j|\text{State } i)$

Existing Systems using HMM for map matching require GPS coordinates as inputs
Start Probabilities

No of Intersections = 4

Start Probability = 1/4

Start states emerge from Intersections within uncertainty range

Stolen item original location
Transition Probabilities (TPs)

We want to assign TPs

From the system’s perspective

States are Road Segments

Error Range of Distance Computation

Gets highest TP since closest to measured distance

Gets TP proportional to their distance from actual measured dist
Observables

Speed of the Car?

Curvatures of Roads?

Sequence of distances between every pair of turns and/or stops experienced by the burglar’s car
Let’s say there are two road segments of identical length. discrimination using location of traffic lights on roads.
Real Life Test

0.958km (1.227)
0.117km (0.095)
0.327km (0.467)
2.604km (2.265)
0.224km (0.255)
0.559km (0.849)
2.158km (1.882)
0.736km (0.549)
2.154km (2.476)
0.193km (0.143)
Real Life Test-Return Path

- University of Memphis: 1.472 km (1.378)
- Park Ave: 0.379 km (0.446)
- Poplar Ave: 0.927 km (0.087)
- Park Ave Shopping Center: 0.687 km (0.897)
- Memphis Medical Center: 6.591 km (6.303)
- Baptist Memorial Hospital-Memphis: 1.984 km (1.784)
- Poplar Ave: 0.086 km (0.112)
Evaluations at Scale

Set up for the evaluations:

- We used Openstreetmap.org GIS data base
- Chose 100 different locations throughout the city as our starting locations
- For each stating location chose 10 different directions for a fixed travelled distance
- Created synthetic paths from the database marking a set of intersections as STOP
Evaluations of Map Matching

Stopping Sequence, Localization at Destination

Max Distance Computation Error

92%

73%

% of Matches

Error Probabilities

25 miles
20 miles
15 miles
10 miles
5 miles
2 miles
Evaluations of Map Matching

Stopping Sequence, Localization at each Turn

Max Distance Computation Error

98 %

93 %

% of Matches

Error Probabilities

- 25 miles
- 20 miles
- 15 miles
- 10 miles
- 5 miles
- 2 miles
Conclusions

- Achieves over 90% accuracy in identifying the path driven by the burglar using inertial estimates
- Immune to Radio Outages
- Life Time prolonged due to filtration by Vibration Dosimeter

Future Work
- Locate apartment or house of the burglar
Road Segment A

Traffic light pinned as it falls within error margin

Road Segment B

RS B pruned as no traffic light matched

Stopping estimates refined by pinning to matched traffic lights

Stops
Both lights within the error range

Branch 1

Branch 2
Evaluations of Map Matching

Cumulative Prob. of correct path in K top weighted paths

91% Chance