GPS Data Collection for Regional Travel Surveys

Timothy Michalowski, GIS Director

t.michalowski@srbi.com

Abt SRBI, 275 Seventh Ave, New York, NY 10001
Introduction

- **Timothy Michalowski, GIS Director, Abt SRBI**
  - 10+ years GIS experience, focus on GIS for Social Research
  - Previously worked at NYC DOT, Puget Sound Regional Council
  - Master of Urban Planning/GIS from University of Illinois (Chicago)

- **Abt SRBI**
  - National Leading survey research firm, founded in 1981
  - 17th largest Research firm in USA (*Honomichl List*)
  - Headquarters in NYC, Offices in DC, Chicago, Boston, Arizona, North Carolina, Florida, Ohio
  - Expertise in 16 practice groups, including Transportation, Social Policy, Market Research, Health, Energy, Elections, GIS, etc.
About Abt SRBI GIS

- Clients Include:
  - Amtrak
  - Girl Scouts of America
  - NYC Economic Development Corporation
  - Port Authority of New York and New Jersey
  - USAID
  - Yum! Brands
  - National Oceanic and Atmospheric Administration (NOAA)
  - U.S. Department of Housing & Urban Development (HUD)
Travel Behavior Surveys

- Metropolitan and Regional Transportation Planning Organizations (MPOs and RTPOs)
- Conducted every 5-15 years
- Used for urban/regional planning
  - Travel demand models
  - Regional capacity and level of service planning
- Survey participants
  - Recruited randomly from general population
  - Self report of travel behaviors
  - Incentives provided for completion
Objectives for Travel Surveys

- What are the **origins/destinations** of individual trips?
- What are the **trip segments**?
- What are the trip **distances/times/speeds**?
- What are the travel **modes**?
- Ensuring high **precision** of data
Traditional Survey Methods v. GPS

**Travel: How did you get to Location 1?**

1. What type(s) of transportation did you use to go to Location 1?  

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd (if needed)</th>
<th>3rd (if needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car, van, truck</td>
<td>Public Bus</td>
<td>Amtrak</td>
</tr>
<tr>
<td>Walk</td>
<td>Light Rail (Hiawatha)</td>
<td>Bicycle</td>
</tr>
<tr>
<td>School Bus</td>
<td>Commuter Rail (Northstar)</td>
<td>Motorcycle/Moped</td>
</tr>
<tr>
<td>Other (specify)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. If you used a bus/Train for this trip, did you use a pass?  □ Yes  □ No → How much did you pay?____

3. If you used car/van/truck or motorcycle/moped for this trip . . .
   A. Were you the . . .?  □ Driver  □ Passenger
   B. Including yourself, how many people were in the vehicle?  1  2  3  4+  
      Including yourself, how many are household members?  1  2  3  4+  
      Which household members were with you?  
      , , , , , , , , 
   C. Was this vehicle from your household?  □ Yes  □ No
   D. Did you pay a toll?  □ Yes  □ No
   E. How much, in total, did you personally pay for parking?  □ Nothing  
      $____ ____ ____ Was the rate . . .?  □ Hourly  □ Daily  □ Monthly  □ Other

**Advantages of GPS:** Route information, lower respondent burden, no data entry, increased data quality and data volume
Abt SRBI GPS Travel Survey
Los Angeles Region

- Southern California Association of Governments (SCAG)
- April 2012 to October 2012
- ~900 households participated
- ~1,800 total GPS units sent out
Abt SRBI GPS Travel Survey
Philadelphia Region

- Delaware Valley Regional Planning Commission (DVRPC)
- August 2012 to April 2013
- ~750 households participated
- ~1,500 GPS units sent out
GPS Loggers vs GPS Smartphones

GPS Data Loggers

- Passive GPS Data Collection
- Usable by all members of general population (~55% of cell phone users have smartphones)
- Collects GPS data every 1 second, batteries last multiple days

GPS Smartphone Application

- Utilization of participants’ current smartphones
- No need to purchase, mail, manage GPS devices
- Customized prompting for additional survey questions

GPS data collection devices result in same core GPS data
Abt SRBI GPS Device

- Lightweight: 2.5 oz
- Records every **1 second** of travel activity
- Passive device – Powers on automatically with movement
- Carried **everywhere**
GPS Deployment

SEND = FedEx

RETURN = USPS

GPS Mail Database
Statistical Tests for GPS Compliance

- **Age** and **household size** are positively correlated with non-compliance in GPS study.
- **Gender** and **regional location** are not correlated with GPS compliance.
- Larger **households size** = more GPS units = greater overall burden.

Incentive of **$25 per GPS unit, not per household**.

Limiting to **4 persons per household > 16 and < 85 years old**.
## Raw GPS Data Output

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>Speed</th>
<th>Course</th>
<th>Sat</th>
<th>HDOP</th>
<th>Altitude</th>
<th>date</th>
<th>time</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>-93.2671</td>
<td>45.09099</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>238</td>
<td>19/8/2011</td>
<td>22:45:01</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09098</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:02</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09098</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:03</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09098</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:04</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09097</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:05</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09097</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:06</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09095</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:07</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09095</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:08</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09094</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:09</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09094</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:10</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09094</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:11</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.0909</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:12</td>
<td>3</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09099</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:13</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09097</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:14</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09096</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:15</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09096</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:16</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09083</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:17</td>
<td>2</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09082</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>237</td>
<td>19/8/2011</td>
<td>22:45:18</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09081</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:19</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09081</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:20</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09078</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:21</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09077</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:22</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09076</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:23</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09075</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>236</td>
<td>19/8/2011</td>
<td>22:45:24</td>
<td>1</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09075</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>239</td>
<td>19/8/2011</td>
<td>22:45:25</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09075</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>239</td>
<td>19/8/2011</td>
<td>22:45:26</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09075</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>239</td>
<td>19/8/2011</td>
<td>22:45:27</td>
<td>0</td>
</tr>
<tr>
<td>-93.2671</td>
<td>45.09075</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>240</td>
<td>19/8/2011</td>
<td>22:45:28</td>
<td>0</td>
</tr>
</tbody>
</table>
GPS Data Before Processing

Distorted waypoints resulting from indoor use

Participant carries GPS units indoors

Extraneous stationary data
Determining Trips from GPS Data

150,000,000+ GPS Points  →  70,000+ Trips
Abt SRBI GPS Processing Model

Step 1:

**Automated Determination of Travel Segments**

Iteration through raw GPS files

Python scripts finding trip starts and ends

Import to geodatabase used for processing

Add fields to a processing table

Output: cleaned feature class of GPS trips per participant
import arcpy, arcgisscripting

# Create the Geoprocessor object
# gp = arcpy.Parameter() [G:gis:project:82250:GIS\GIS\GIS_PROJECTS\S3S_SCAG\PYTHON\GPS\data\Python\GPS\data\Python]
# OUTPUT_FOLDER = (r'G:\GIS\PROJEN\S3S_SCAG\PYTHON\GPS\data\Python\GPS\data\Python')
# Output_Layer = "points_layer"
# ENVIRONMENT = [r'G:\GIS\PROJEN\S3S_SCAG\PYTHON\GPS\data\Python\GPS\data\Python']
# arcpy.environ
# CURSOR = "points_layer"
# arcpy.env
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
# CURSOR = "points_layer"
Trip Segment Intervals

Travel Speed (mph)

Travel Time (seconds)

< 120 second gaps = Same trip segment (stop light)
> 120 second gap = New trip segment
Step 2:

Manual Review of Trip Segments

- Zooms the map sequentially to each flagged trip
- User accepts or rejects trip segments
- Layer symbology updated to show approval status of trips

```python
import arcpy

MXD = arcpy.mapping.MapDocument("Current")
listLayers = arcpy.mapping.ListLayers(MXD)

dataframe = arcpy.mapping.ListDataFrames(MXD, "Layers")[0]

pointsLayer = arcpy.mapping.ListLayers(MXD, "GFS_POINTS")

allFrames = arcpy.mapping.ListDataFrames(MXD)

for dataframe in allFrames:
    MXD.activeView = dataframe
    for points in pointsLayer:
        arcpy.ApplySymbologyFromLayer_management(points, r"C:\GIS\GIS_5385_SCAG"

arcpy.RefreshActiveView()
```
Step 3:

Trip Speed Calculations & Final Products

• Uses the datetime module in python to calculate total trip time

• Calculates length of trip line to determine trip distance

• Divides distance by time to generate trip speed

```python
for x in rows:
    if x.getValue['final'] == "START" and x.getValue('accepted') == 1:
        startlist.append(x.getValue('date'))
        startlist.append(x.getValue('time'))
    elif x.getValue('final') == "END":
        endlist.append(x.getValue('date'))
        endlist.append(x.getValue('time'))

del x, rows

startlist = []
endlist = []

while len(startlist) > 0 and len(endlist) > 0:
    ds = startlist[0]
    ts = startlist[1]
    tst = ts.time()
    de = endlist[0]
    te = endlist[1]
    tet = te.time()

    startlist.pop(0)
    startlist.pop(0)
    endlist.pop(0)
    endlist.pop(0)
```

Numbered Trip Segments, Categorical Symbology by Trip Number

Trip Segments Symbolized by Speed

Higher average speed
Final Products

- Route information in line and point format
- Origin and destination points for each trip
- Removal of stationary non-trip data
- Improved accuracy of trip distance, time, and speed calculations compared to diaries
GPS Final Products

Raw GPS Data → Cleaned Points → Output Trip Lines
GPS Final Products
GPS Travel Survey Challenges

- Determining **mode** of travel
  - Combination of speed, routes
  - Bicycling can mimic vehicles in traffic
  - Studying collected travel speed patterns
- Ensuring new trips are new trips
  - LA traffic > 2 minutes
  - Goal: limiting manual verification
- Capturing tunnel travel (subway)
GPS Trave Survey Challenges

- Capturing tunnel travel (subway)

**A. Line Dataset**

**B. Final Point Deliverables**
GPS Travel Survey Conclusions

- **Successful data collection method**
  - 200+ million GPS points collected
  - 30,000+ days of travel information collected

- **Lowers respondent burden**
  - Higher response rates with GPS compared to travel diaries
  - No filling out of lengthy forms

- **GPS Loggers work for now, smartphones are next step**
  - GPS loggers and Smartphones used in collaboration to reach all populations

- **More robust, accurate data for planning**
  - Route information
  - Eliminates data entry errors
  - Output integrates with travel demand models

- **ESRI products provide the necessary tools**
  - SQL Server 64 bit for all data storage
  - ModelBuilder, Python
  - ArcMap for Review (developing a mapping API)
Thank you

Timothy Michalowski
Abt SRBI
GIS Director
275 7th Ave, Suite 2700
New York, NY 10001

646-486-8404

t.michalowski@srbi.com