Macroscopes/Mobiscopes

- Macroscopes
  - More resourceful sensors than mote-like devices
    - Cameras, Microphones, Mobile phones, GPS etc..
    - Typically not battery operated
  - Larger geographical scope

- Mobiscopes
  - Same as above, but mobile
  - Many advantages over static sensors:
    - Can cover more ground
    - Useful with expensive sensors
  - Nearly unprecedented and constantly growing reach
Macroscopes/Mobiscopes

- Enable many applications
- Inrix
- Traffic cameras (tracking, surveillance etc)
- Imaging the world (Google Earth)
  - Incredible number of pictures already available for almost anywhere
- Public health epidemiological studies
  - Imagine having live data for every person at any time
  - Most cellphones of future will come with GPS
  - Information already being used for traffic monitoring
- Automated image capture to provide feedback on diet, exercise etc..
  - MyLifeBits (Gordon Bell)
- IT for developing countries
A network of web-cameras
- Parking spot monitoring, bus monitoring

The images were processed at the sensors to extract information

The information presented to the users as an XML document
- Querying using XPATH

Interesting distributed query processing issues

Image processing turned out to be much harder than expected
Federating/visualizing sensor information

- SensorMap
- Goal: Make it easy to publish sensor data
- Google Earth similar, but not real-time
“Mobiscopes for Human Spaces”; IEEE Pervasive Computing, 2007 (Sam Madden a co-author)

A federation of distributed mobile sensors

- Achieves high-density sampling because of mobility
- Not always possible to instrument precisely
- Sensors may be very expensive
- Naturally mobile sensors (people with cameras)
Mobiscopes: Challenges

- Application adaptation
  - Must be designed to work with intermittent connectivity

- Actuated mobility
  - May be able to direct sensors to locations
  - Hard “value of information” problems (possible class projects)

- Opportunistic connectivity
  - How to quickly identify? How to route?

- Prioritization
  - Because of limited connectivity and too much data

- Heterogeneity
  - Not specific to mobiscopes; more generally an issue with the world wide sensor web

- Data Privacy ???
Sensors in cars monitoring:
- GPS, OBD (II), Images, Wireless network connectivity
- Can add more sensors easily

Goal: Build a general platform that
- Provides a simple programming interface
- Handles intermittent and variable network connectivity
- Handles data heterogeneity

Applications:
- Environmental monitoring, Civil infrastructure monitoring
- Automotive diagnostics, Geo-imaging, DA ta muling
- …
CarTel: Architecture

- **ICEDB Server**
  - Answers local snapshot queries
  - Logs continuous query results
  - Prioritizes data

- **Open Wireless Access Point**

- **Portal**

- **Clients**
  - **CafNet**
    - Delay-tolerant relay via 802.11, Bluetooth, etc.
  - **User’s Wireless Access Point**
  - **ICEDB Remote**
    - Adapters log GPS, Wi-Fi, OBD, camera data

Data sent via prioritized continuous queries.
A server component, and a client component (remote)

Two types of queries

**Continuous** (Push) queries
- Essentially specifies the sampling rates (but could be more complex)
- No results are sent to the users or other applications
- Controls the “acquisition” of data

**Snapshot** (Pull) queries
- Applications/users can ask queries at any time against the collected data

Not a “streaming” database system as we use the word
- Even at local sensor nodes, the data stored in the local DB first

Use an extended version of SQL for specifying the continuous queries
Queries specify priorities for the data
  - Build this in from the beginning
Inter-query and intra-query priorities
Two novel ideas:
1. A ranking function that operates on the entire result at once
   - Typically ranking functions operate on a per tuple basis
   - In some sense, the “value” of a tuple is dependent on other tuples (another possible project)
2. Server feedback
   - Send a summary to the server
   - Server sends back its perceived priorities (on groups)
Question: How expressive is their approach?
- There is a trade-off between making things easy to specify (declarative) and the expressive power.
- Their proposed interface is limited to providing the feedback at “group” level.
- Can you do better?
- Do you need to?
CarTel: CafNet

- Handles the network connectivity and message delivery
- Deals with variable network types
  - Maybe Wi-Fi, Bluetooth, possibly Cellular etc..
- Allows for dynamically changing priorities
  - Queue packets at the transport layer
  - Doesn’t work when “priorities” are so important
  - Because of this, somewhat more involved
CarTel: CafNet - Stack

![Diagram of the CafNet communication stack]

**App 1**  ...  **App N**

**Transport Layer**
- registers data to be transmitted
- delivers incoming data
- requests data from the application
- notifies application of successful delivery

**Network Layer**
- notifies transport layer of free buffers
- schedules data for transmission
- selects routes
- buffers data for transmission

**Mule Adaptation Layer**
- provides uniform neighbor discovery

**Device Driver**  ...  **Device Driver**

**Figure 3. The CafNet communication stack.**
CarTel: CafNet

- Lowest layer calls back into the upper layers when connectivity
- The upper layer can decide at that time what to do

Problem:
- Takes too much time
- Connectivity is shorter than that

Solution:
- Some limited buffering
- Lose some control over the priorities
Some related work

Delay Tolerant Networking (DTN)
- An important research area
- Especially in developing countries (although cellular penetration really good)
- The idea of using “data mules” quite common

Ranking in Database Systems
- Similar to priorities
- Much work on this