MMI 2: Mobile Human-Computer Interaction

Location & Context

Prof. Dr. Michael Rohs

michael.rohs@ifi.lmu.de

Mobile Interaction Lab, LMU München

Lectures

#	Date	Topic
1	19.10.2011	Introduction to Mobile Interaction, Mobile Device Platforms
2	26.10.2011	History of Mobile Interaction, Mobile Device Platforms
3	2.11.2011	Mobile Input and Output Technologies
4	9.11.2011	Mobile Input and Output Technologies, Mobile Device Platforms
5	16.11.2011	Mobile Communication
6	23.11.2011	Location and Context
7	30.11.2011	Mobile Interaction Design Process and Prototyping
8	7.12.2011	Evaluation of Mobile Applications
9	14.12.2011	Visualization and Interaction Techniques for Small Displays
10	21.12.2011	Mobile Devices and Interactive Surfaces
11	11.1.2012	Camera-Based Mobile Interaction 1
12	18.1.2012	Camera-Based Mobile Interaction 2
13	25.1.2012	Sensor-Based Mobile Interaction 1
14	1.2.2012	Sensor-Based Mobile Interaction 2
15	8.2.2012	Exam

Review

- Name wireless communication technologies
- What technology for transmitting sensor data?
- What technology requirements for voice calls?
- Design goals of Bluetooth?
- What is a Piconet? Device roles?
- How are connections established in Bluetooth?
- How is power saved in Bluetooth?
- What is SDP?
- What are examples of cell-based systems?

Preview

- Context
- Describing and deriving location
- Location systems
- Application examples

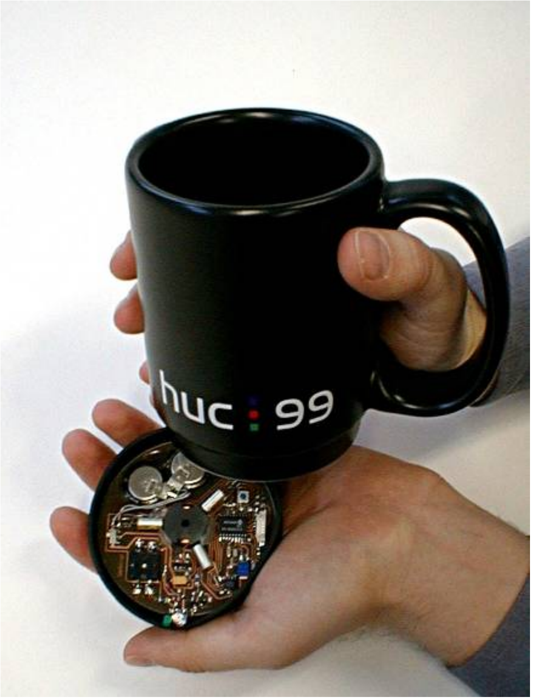
- Android Location & Map APIs
- Android Media Framework

CONTEXT

Characteristics of Context

- Context
  - Where you are, who you are with, what resources are nearby
  - Information about the user, the user's environment, the device's context of use
- User's context changes rapidly when mobile
  - User interacts with many devices, people, objects, and places
- Context-aware applications
  - Capture and retrieve context information
  - Adapt to the user's context
  - Reduce need for explicit user input
  - Are better integrated with user's environment and activity

Active Artifacts



- Determine activity where it occurs
- Add “self perception” to everyday things
- Communicate their own state
- The artifact digitally “supports” its own applications
- Example: MediaCup
  - http://mediacup.teco.edu
- Exercise: Assume MediaCup should be able to discriminate
  - Informal meeting
  - Presentation Coffee break
  - Working alone

  → What sensors? How to represent the situation?

Defining Context

- Context-aware computing (Schilit and Theimer, 1994)

  Software that “adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time”

- Context (Dey, 2001)

  “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves, and by extension, the environment the user and applications are embedded in.”

Context Sources

- Current location
- Location history
- Orientation
- Speed
- Time of day
- Day of week
- Illumination
- Noise level
- Temperature

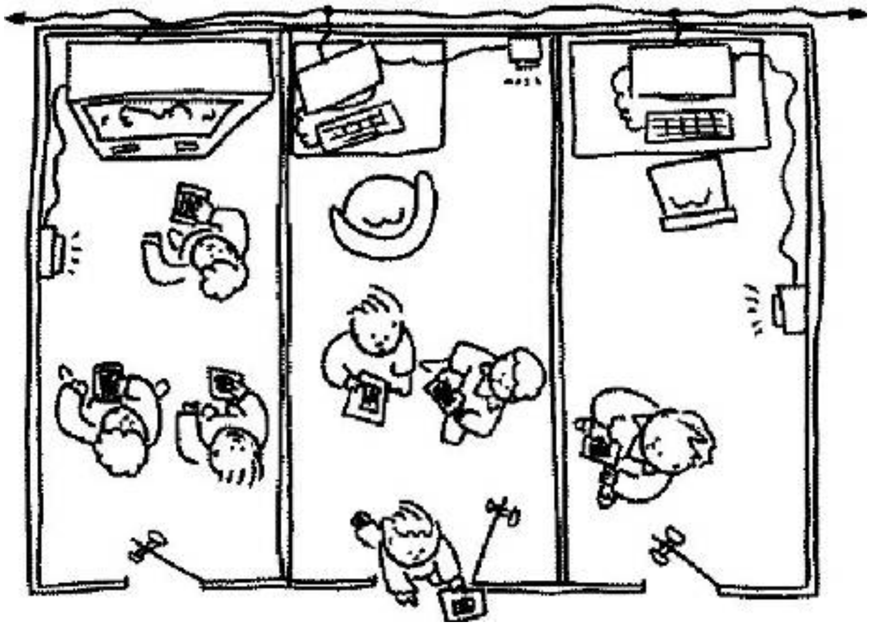
- Network availability
- Network bandwidth
- Remaining battery life
- Device movements
- Dialogue history
- User’s activity & schedule
- User’s mood
- Group constellation
- Number of people around

Location and Context

- Location is an important part of the user’s context
- Locations have a context
  - Noise level and its fluctuations
  - Light level and its fluctuations
  - Number of people in the space
  - Relationship between people present
  - Social protocol at that location
  - Activity for which the location is designed

Techniques in Context-Aware Computing
(Schilit, Adams, Want: Context-Aware Computing Applications. 1994)

- Proximate selection
  - E.g., list nearest printer first
- Automatic contextual reconfiguration
  - E.g., share nearby electronic whiteboard automatically
- Contextual information and commands
  - Commands with different meanings in different contexts
  - Command only in certain contexts
- Context-triggered actions
  - Condition-action rules



Source: Schilit et al. 1994

Context-Triggered Actions

- Simple if-then-rules, similar to Unix CRON-Demon:
  - Coffee Kitchen arriving „play -v 50 /sounds/rooster.au
  - Schilit * attention „emacs -display $NEARESTHOST:0.0“

- Contextual reminders: information is displayed under certain conditions. Example:

$DATE=„after April 15“
AND $TIME=„after 10“
AND $room=„35-2-200“
AND $WITH-USER=Adams“
AND Color($DISPLAY)=„true“

Difficult Problems in Context-Aware Systems

- How to abstract relevant (higher level) context from low level sensor data
  - How do derive user's intent and situation?
  - Sensors → features → context → intent/situation
  - How to deal with uncertainty in context recognition?
- How to model and exchange context data?
  - Going beyond basic sensor data mining requires AI techniques, knowledge representation (ontologies, taxonomies)
- How to apply the obtained context information
  - Implicit vs. explicit control of systems
- Intelligibility
  - Helping the user to understand system actions
  - Proactivity vs. losing control

Usability Risks for Mobile Context-Aware Applications (Dey, Häkkilä, 2006)

- Uncertainty in context recognition
- Information overflow
- Lack of user control
- Application complexity
- Privacy violations
- Subjective understanding of context attributes
- Lack of common agreed ontologies
- Imbalance between automatic and user-initiated actions
- Poor interoperability

Dey, Häkkilä: Context-Awareness and Mobile Devices. Handbook of Research on User Interface Design and Evaluation for Mobile Technology. 2008.

Design Guidelines for Mobile Context-Aware Applications (Häkkilä, Mäntyjärvi, 2006)

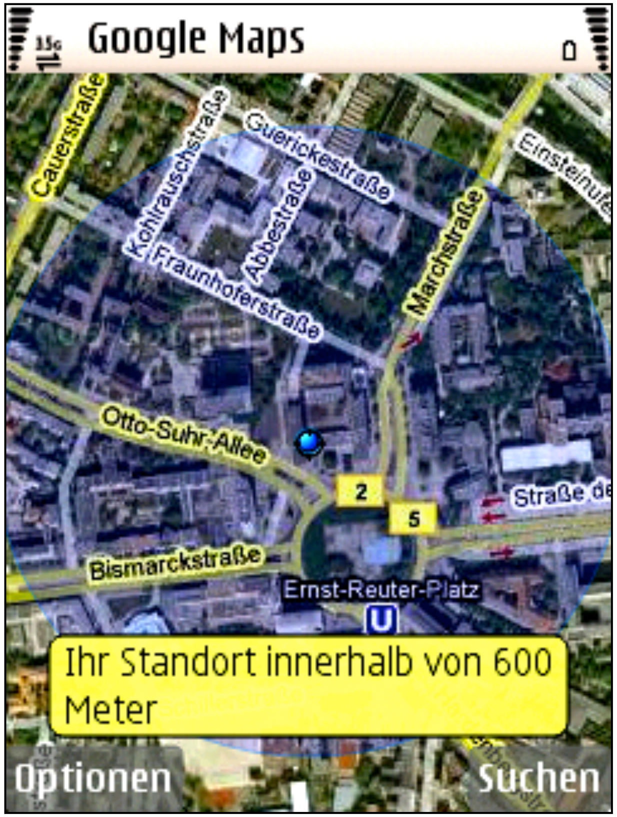
- Select appropriate level of automation
  - Depending on level of uncertainty
- Ensure user control
- Avoid unnecessary interruptions
  - Intrusive, distract, but can have high value
- Avoid information overflow
- Appropriate visibility level of system status
- Personalization for individual needs
- Secure user’s privacy
- Take into account the impact of social context

Häkkilä, Mäntyjärvi: Design Guidelines for Context-Aware Mobile Applications. Proceedings of Mobility’06, ACM Press, 2006.

LOCATION

Aspects of Location Information

- Position vs. place
  - 52N 13E vs. university main building
- Absolute vs. relative
  - 52N 13E vs. 10 km west of where I am
- Representation of uncertainty
  - A few km vs. room level vs. a few cm
- Indoor vs. outdoor
  - Elevation / floor number difficult
- Privacy model
  - Self-positioning vs. infrastructure-based



Ihr Standort innerhalb von 600 Meter

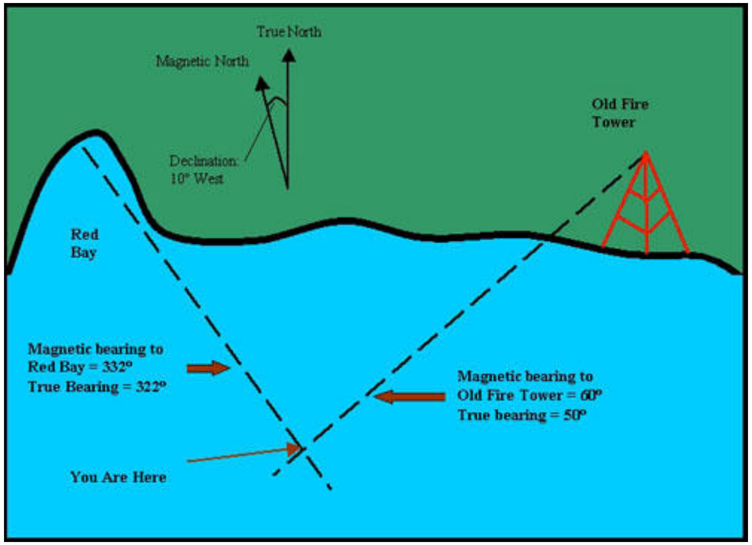
Blue area exposes uncertainty

Some Location Technologies

- GPS
- Wi-Fi access points
- GSM cell
- Ultrasound (time of flight) + radio signal
- Camera: Visual recognition
- Floor pressure
- Signal strength
- Laser range-finding
- Proximity and physical contact (RFID, NFC)

Location Methods

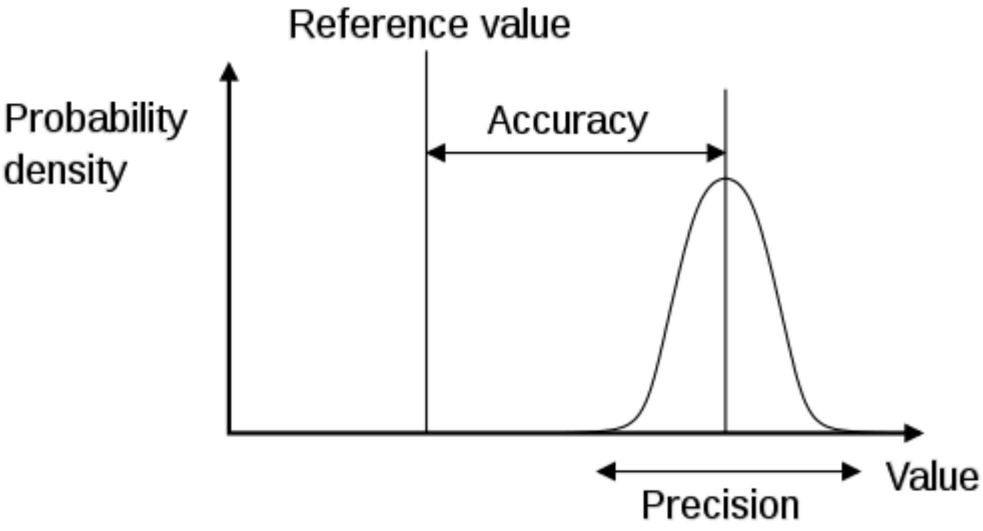
- Trilateration
  - Distance to 3 reference points yields 2D position (4 in 3D)
  - Measuring distance
    - Time of flight (3ns/m for EM, 3ms/m for sound)
    - Signal strength (the stronger the signal, the closer we are)
  - Sources of error
    - Resolution, reflections, multi-path effects
- Triangulation
  - Measure angles instead of distances
  - Sources of error as above
- Fingerprinting
  - Correlation with past observations
  - Database of environment properties
    - Vision, radio signals, etc.



DERIVING LOCATION

Challenges of Deriving Location

- Uncertainty
  - Accuracy, precision
- Cost
  - Money, energy
- Responsiveness
  - Time to result
- Ubiquity
  - Coverage, indoors, outdoors



High accuracy, low precision

Low accuracy, high precision

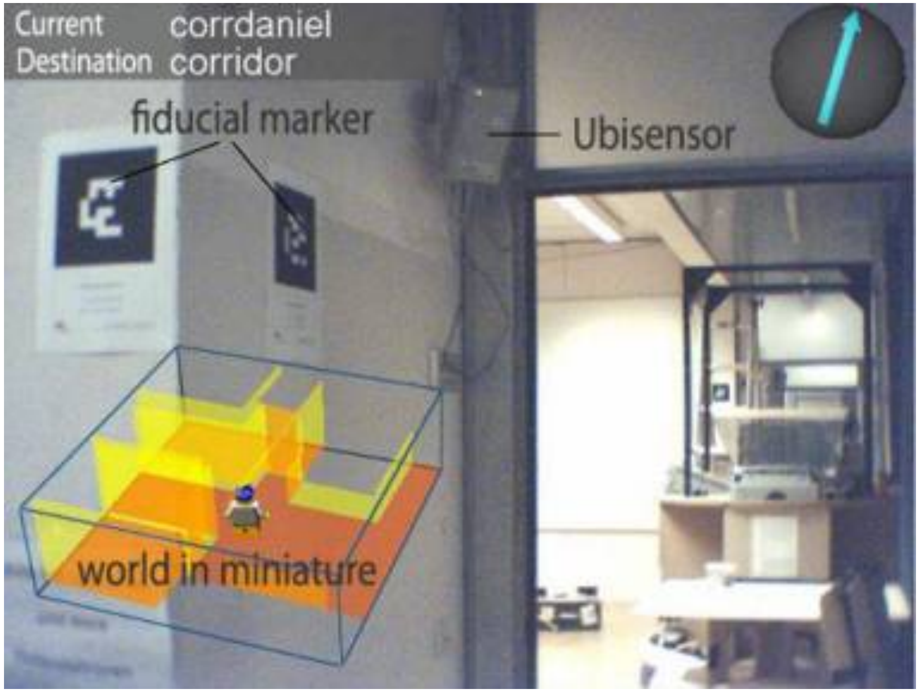
Odometry and Inertial Systems

Odometry: Greek words hodos (“travel”, “journey”) and metron (“measure”)

- Measure change in linear/angular position, velocity, or acceleration to estimate position
  - Example: Measure wheel iterations during vehicle navigation
- Single or double integration accumulates error
  - For short distances/times only

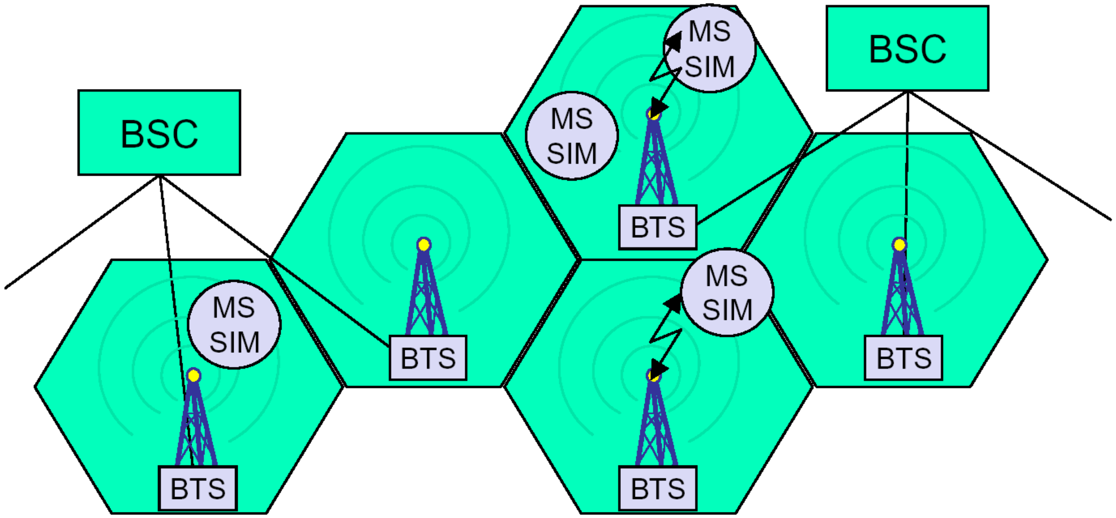
Computer Vision-Based Tracking

- Try to detect
  - Objects directly (persons, landmarks)
  - Textures (e.g. floor texture)
  - Fiducials (2D-Barcodes)
- Problems
  - Image processing expensive
  - Often not robust enough



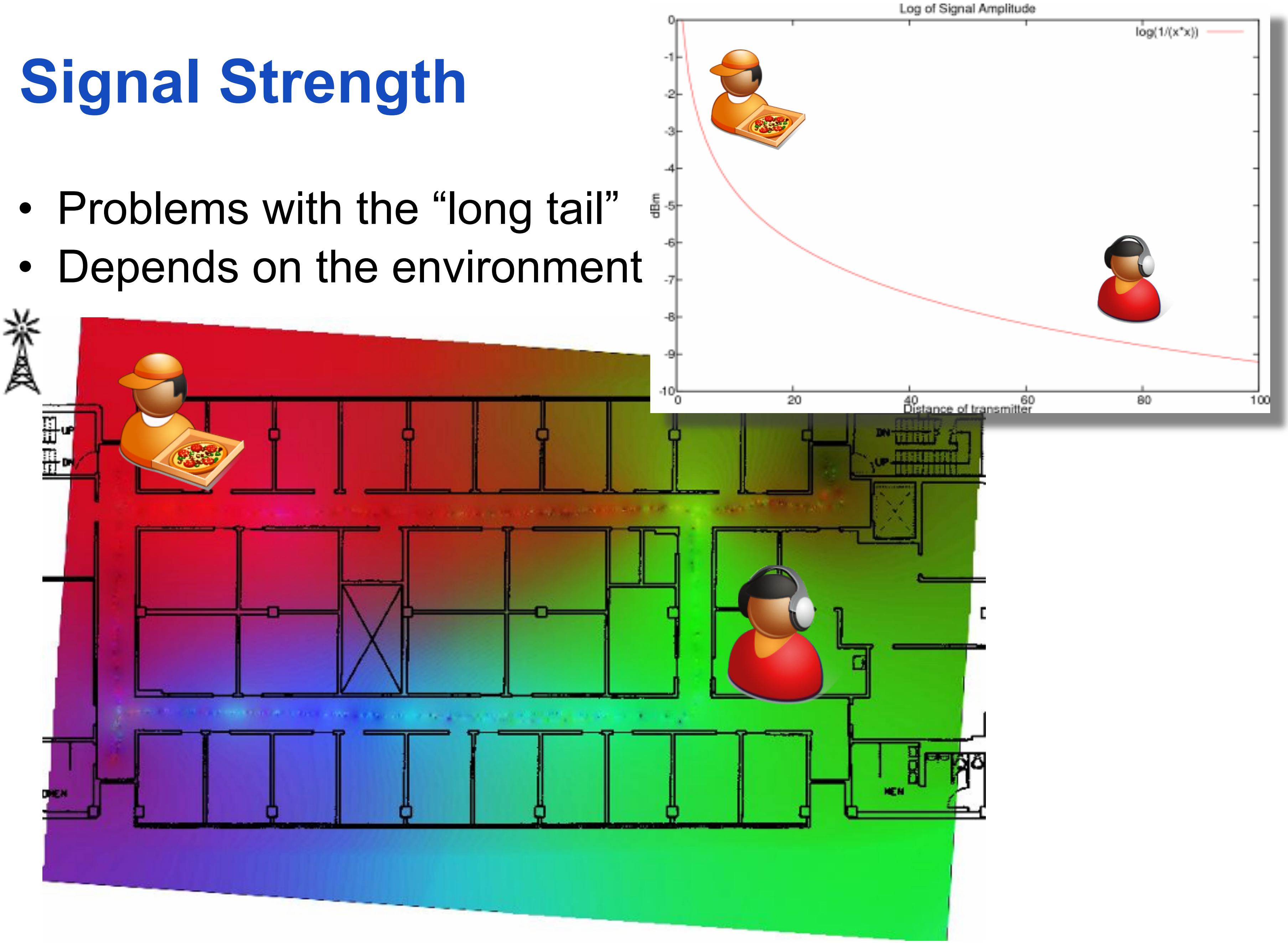
Cell-based Positioning

- Basic GSM-based positioning
  - Cell-ID: 300m (city) to 20km (rural areas)
- More precise GSM-based positioning with
  - Sectored cells (angle of arrival)
  - Multiple base stations (time of arrival, signal strength)
- Only Cell-ID positioning always provided



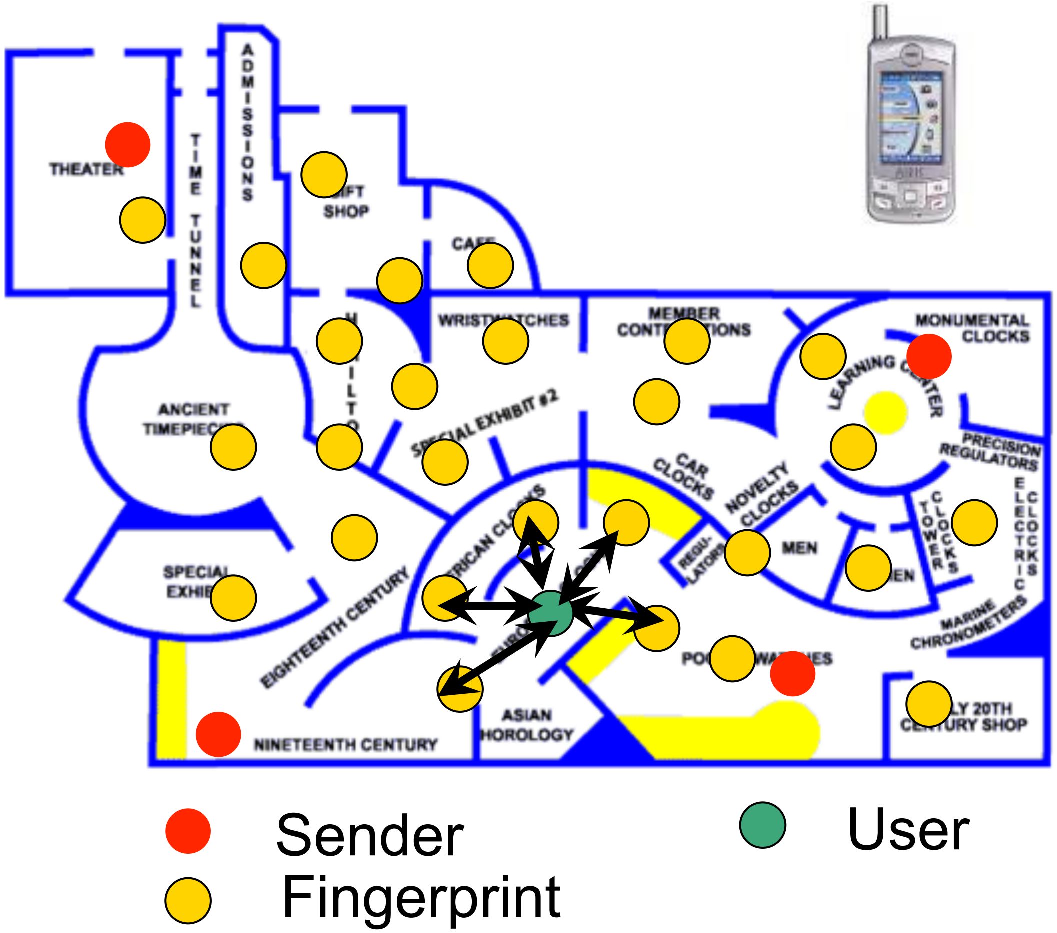
Signal Strength

- Problems with the “long tail”
- Depends on the environment



Fingerprints

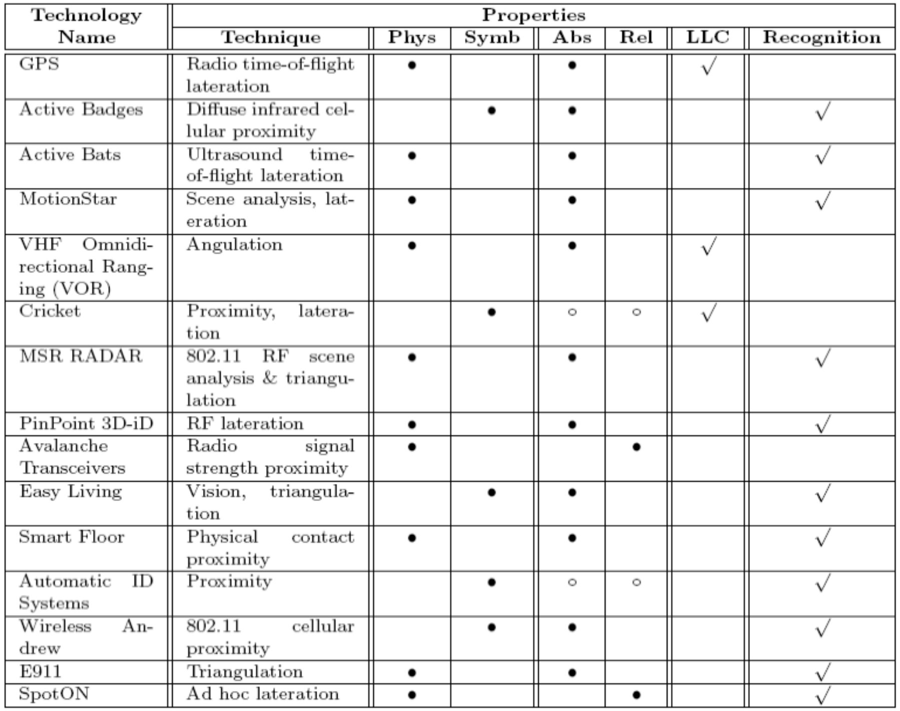
- Measuring of signal strengths at lots of locations (fingerprints)
- Determine position by measuring differences to fingerprints
  - Accuracy: 2-5 m



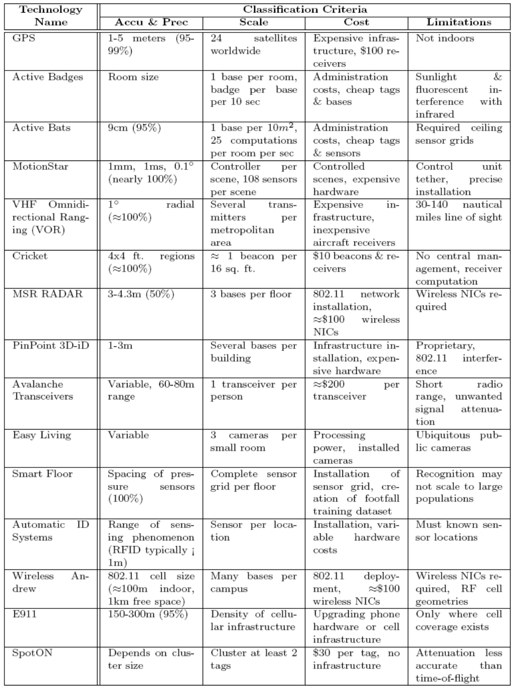
LOCATION SYSTEMS

Taxonomy of Location Systems

- Location Systems for Ubiquitous Computing
  - Jeffrey Hightower and Gaetano Borriello
  - www.cse.iitb.ac.in/~varsha/allpapers/wireless/locationDet.pdf

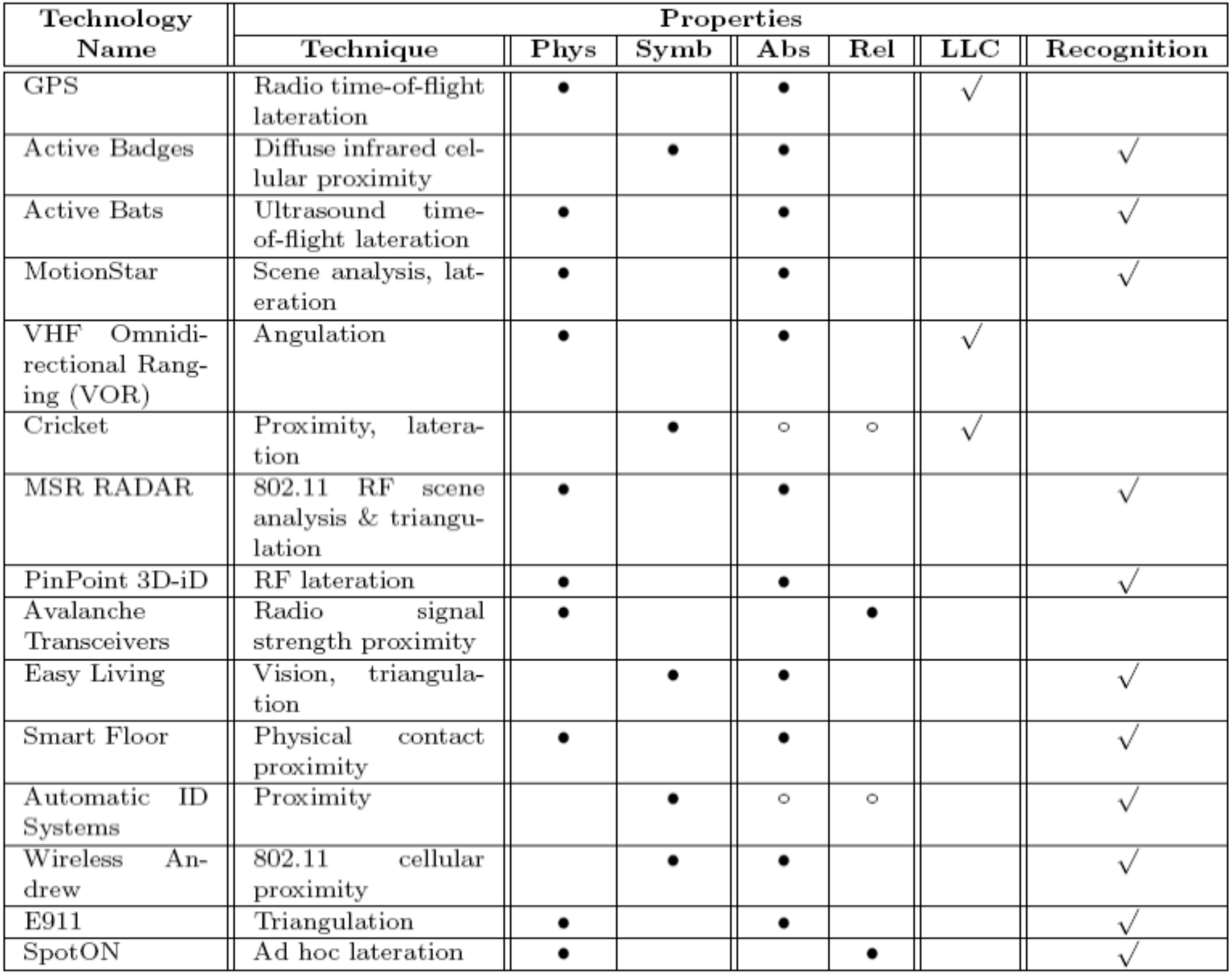


Technology Name	Properties						
	Technique	Phys	Symb	Abs	Rel	LLC	Recognition
GPS	Radio time-of-flight lateration	●		●		√	
Active Badges	Diffuse infrared cellular proximity		●	●			√
Active Bats	Ultrasound time-of-flight lateration	●		●			√
MotionStar	Scene analysis, lateration	●		●			√
VHF Omnidirectional Ranging (VOR)	Angulation	●		●		√	
Cricket	Proximity, lateration		●	○	○	√	
MSR RADAR	802.11 RF scene analysis & triangulation	●		●			√
PinPoint 3D-iD	RF lateration	●		●			√
Avalanche Transceivers	Radio signal strength proximity	●			●		
Easy Living	Vision, triangulation		●	●			√
Smart Floor	Physical contact proximity	●		●			√
Automatic ID Systems	Proximity		●	○	○		√
Wireless Andrew	802.11 cellular proximity		●	●			√
E911	Triangulation	●		●			√
SpotON	Ad hoc lateration	●			●		√



Technology Name	Classification Criteria			
	Accu & Prec	Scale	Cost	Limitations
GPS	1-5 meters (95-99%)	24 satellites worldwide	Expensive infrastructure, $100 receivers	Not indoors
Active Badges	Room size	1 base per room, badge per base per 10 sec	Administration costs, cheap tags & bases	Sunlight & fluorescent interference with infrared
Active Bats	9cm (95%)	1 base per $10m^2$, 25 computations per room per sec	Administration costs, cheap tags & sensors	Required ceiling sensor grids
MotionStar	1mm, 1ms, 0.1° (nearly 100%)	Controller per scene, 108 sensors per scene	Controlled scenes, expensive hardware	Control unit tether, precise installation
VHF Omnidirectional Ranging (VOR)	1° radial (≈100%)	Several transmitters per metropolitan area	Expensive infrastructure, inexpensive aircraft receivers	30-140 nautical miles line of sight
Cricket	4x4 ft. regions (≈100%)	≈ 1 beacon per 16 sq. ft.	$10 beacons & receivers	No central management, receiver computation
MSR RADAR	3-4.3m (50%)	3 bases per floor	802.11 network installation, ≈$100 wireless NICs	Wireless NICs required
PinPoint 3D-iD	1-3m	Several bases per building	Infrastructure installation, expensive hardware	Proprietary, 802.11 interference
Avalanche Transceivers	Variable, 60-80m range	1 transceiver per person	≈$200 per transceiver	Short radio range, unwanted signal attenuation
Easy Living	Variable	3 cameras per small room	Processing power, installed cameras	Ubiquitous public cameras
Smart Floor	Spacing of pressure sensors (100%)	Complete sensor grid per floor	Installation of sensor grid, creation of footfall training dataset	Recognition may not scale to large populations
Automatic ID Systems	Range of sensing phenomenon (RFID typically ¡ 1m)	Sensor per location	Installation, variable hardware costs	Must known sensor locations
Wireless Andrew	802.11 cell size (≈100m indoor, 1km free space)	Many bases per campus	802.11 deployment, ≈$100 wireless NICs	Wireless NICs required, RF cell geometries
E911	150-300m (95%)	Density of cellular infrastructure	Upgrading phone hardware or cell infrastructure	Only where cell coverage exists
SpotON	Depends on cluster size	Cluster at least 2 tags	$30 per tag, no infrastructure	Attenuation less accurate than time-of-flight

Table 2: Location system classification criteria.



Technology Name	Properties						
	Technique	Phys	Symb	Abs	Rel	LLC	Recognition
GPS	Radio time-of-flight lateration	●		●		√	
Active Badges	Diffuse infrared cellular proximity		●	●			√
Active Bats	Ultrasound time-of-flight lateration	●		●			√
MotionStar	Scene analysis, lateration	●		●			√
VHF Omnidirectional Ranging (VOR)	Angulation	●		●		√	
Cricket	Proximity, lateration		●	○	○	√	
MSR RADAR	802.11 RF scene analysis & triangulation	●		●			√
PinPoint 3D-iD	RF lateration	●		●			√
Avalanche Transceivers	Radio signal strength proximity	●			●		
Easy Living	Vision, triangulation		●	●			√
Smart Floor	Physical contact proximity	●		●			√
Automatic ID Systems	Proximity		●	○	○		√
Wireless Andrew	802.11 cellular proximity		●	●			√
E911	Triangulation	●		●			√
SpotON	Ad hoc lateration	●			●		√

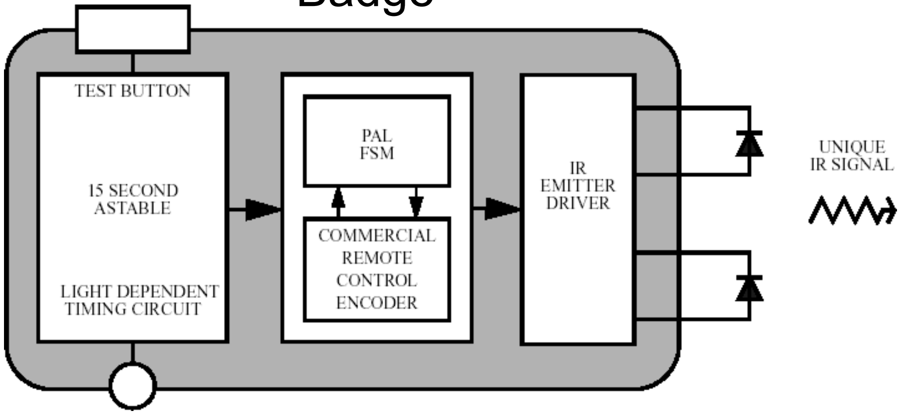
The Active Badge System (1990)

- Olivetti / AT&T
  - Schilit, Hopper, Harter, et al.
  - http://www.cl.cam.ac.uk/research/dtg/attarchive/ab.html
- Teleport
  - Redirect screen output from “home” computer to nearby computer
- Phone forwarding
  - Automatically forward phone calls to nearest phone

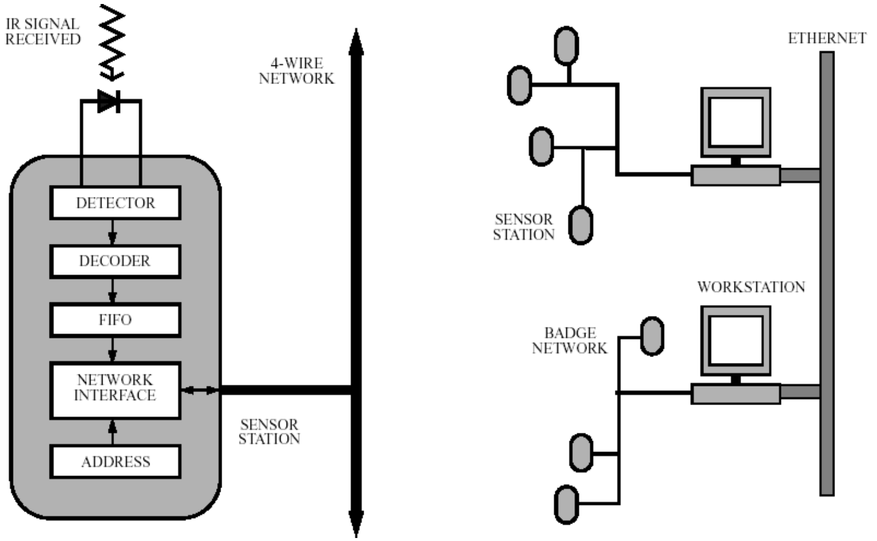
Active Badges: Technology

- Badges worn by users
  - Emit unique IR signals
  - 1 signal every 15 sec
- IR sensors distributed in building
  - Room level accuracy
- Central server scans for “badge sightings”

Badge

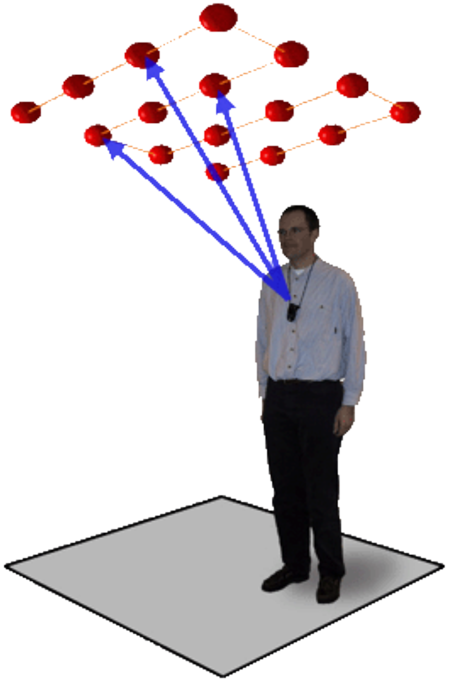


IR Signal



Active Bats Ultrasonic Location System

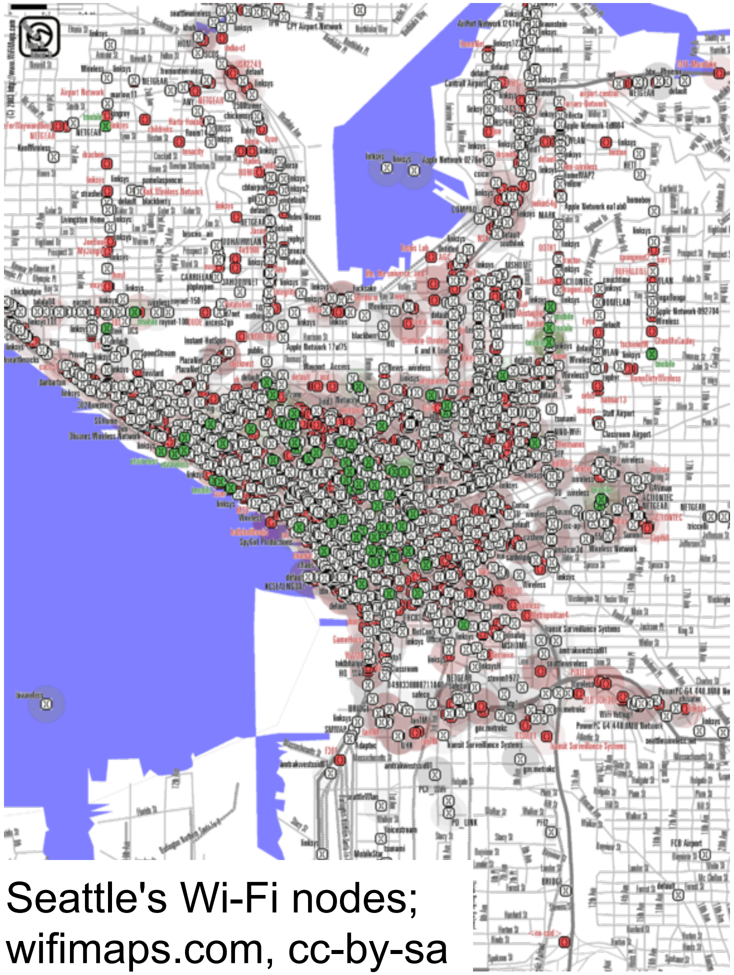
- Fine-grained 3D position and orientation tracking
- Trilateration: position finding by measuring distances
- Ultrasound signal synchronized with RF signal
- 2-5 cm accuracy, expensive, 1 receiver/m^2



http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/
http://www.cl.cam.ac.uk/research/dtg/attarchive/spirit/

Place Lab Location System

- Uses existing RF sources: Wi-Fi, GSM, Bluetooth
- "War-driving" databases
  - Walk/drive around and associate Wi-Fi and GSM signals with current GPS position
  - Explore positions of stations
  - Difficult to keep current
- Accuracy
  - 15-30m (in dense areas) for Wi-Fi
  - 150-300m for GSM
- Lots of projects
  - http://www.placelab.org/projects/

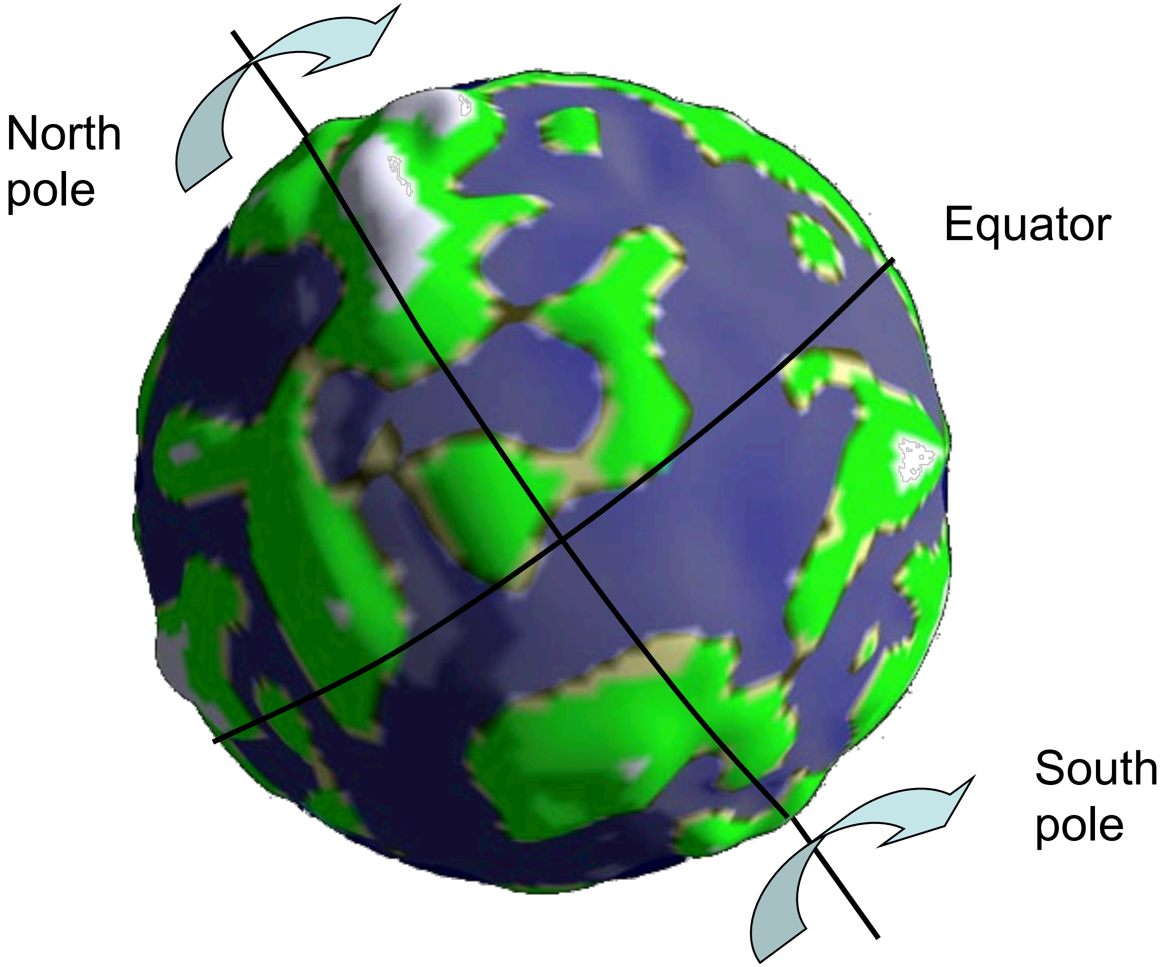


Seattle's Wi-Fi nodes; wifimaps.com, cc-by-sa

GLOBAL LOCATION

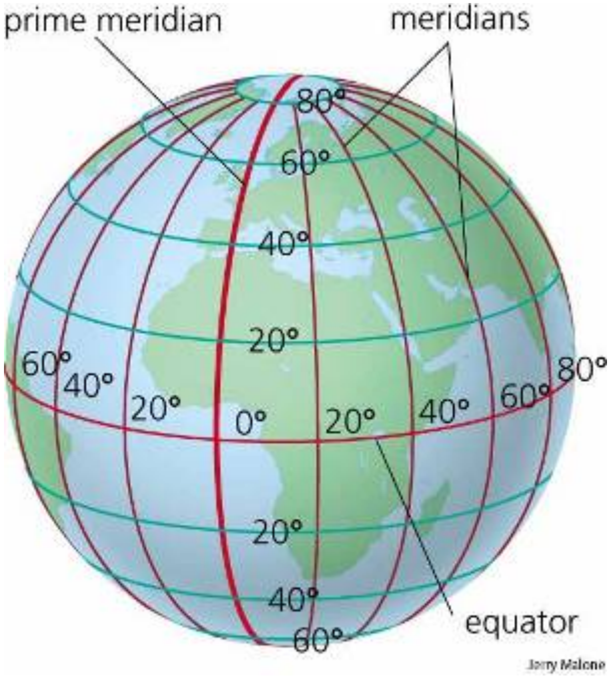
Global Reference System?

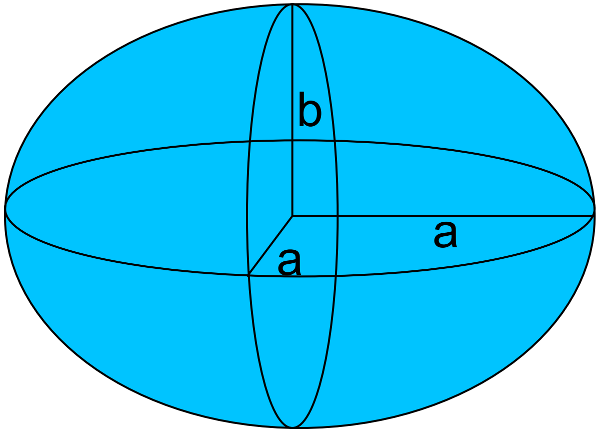
- How to denote locations on a sphere?

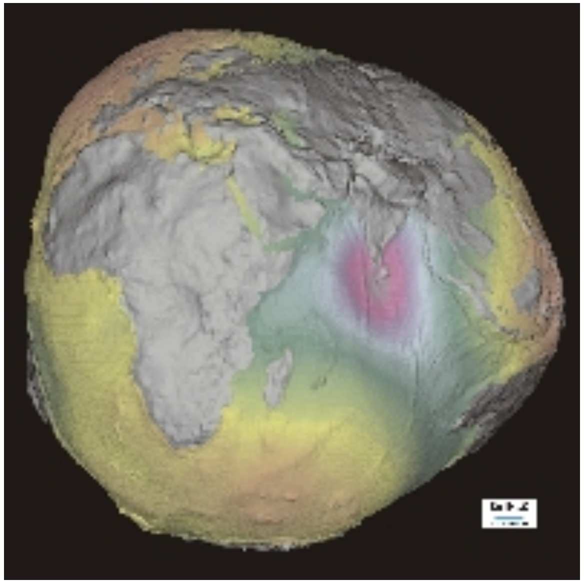


Location Reference System: Coordinates ≠ Coordinates

- Treat the earth as a sphere, ellipsoid or geoid



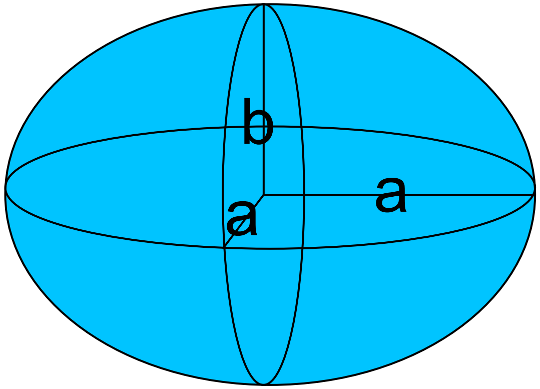




WGS 84 – A Global Reference System

- The World Geodetic System defines a reference frame for the earth, for use in geodesy and navigation.

- Using a special ellipsoid: WGS 84
  - a= 6 378 137.000 m
  - b= 6 356 752.314 m
  - Inverse flattening: f = 298.257 := (a-b) / a

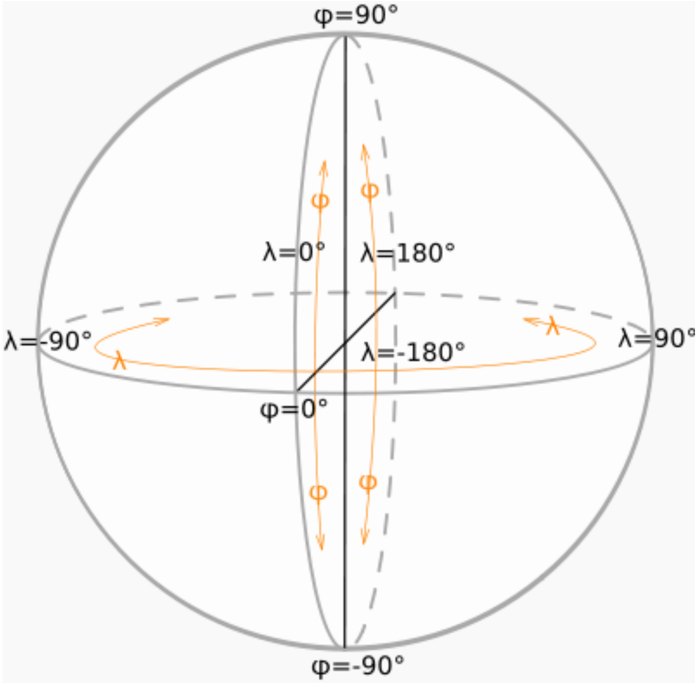


- Geocentric and globally consistent within ±1 m
  - Latitudes, longitudes, altitudes

WGS 84 – A Global Reference System

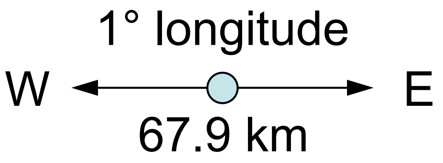
Laser projected from observatory marking Prime Meridian line

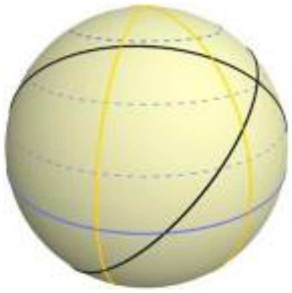
- Longitudes
  - λ = 0°: 102.5m east of the Greenwich Prime Meridian (Greenwich Royal Observatory)
  - λ > 0°: east
  - λ < 0°: west
- Latitudes
  - φ = 0°: equator
  - φ = 90°: north pole
  - φ = -90°: south pole
- 1714 “longitude act”: UK government offered £20000 reward for method to determine longitude

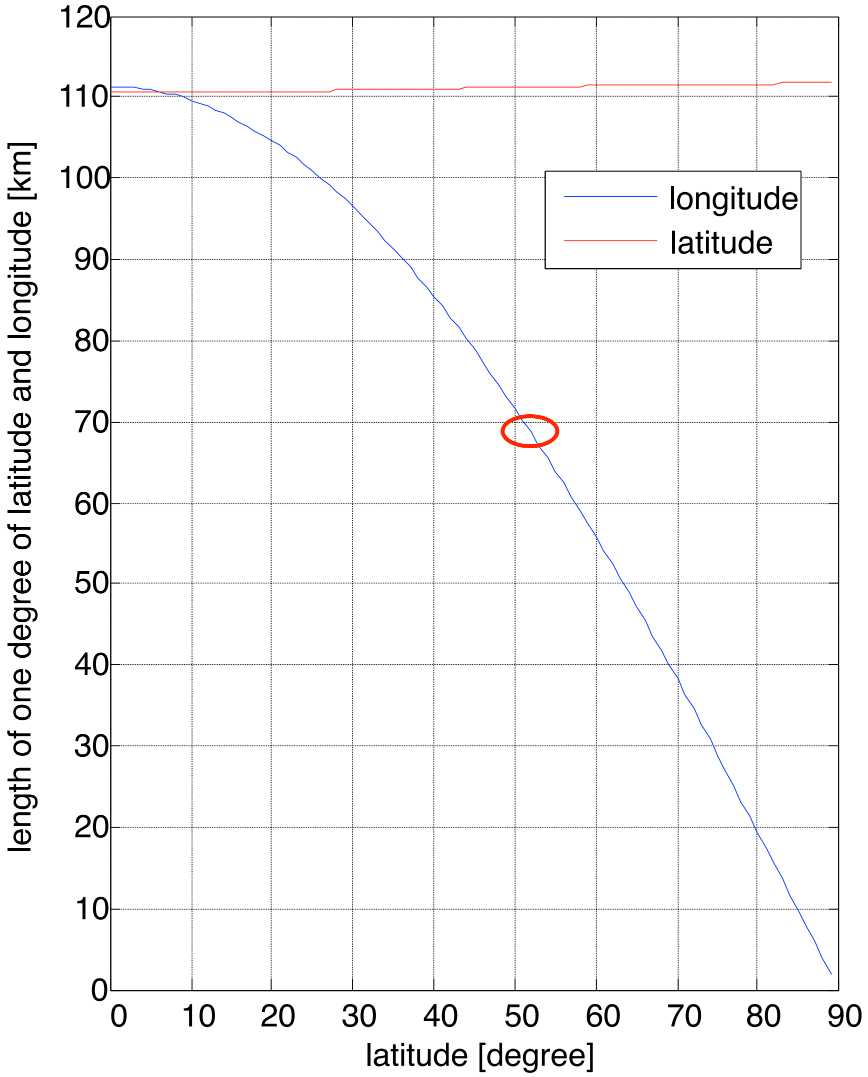


Length of 1° Longitude in Meters

- At equator (lat. = 0°)
  - 1° lon. = 1/360 of the length of the equator = 111.3 km
- At North Pole (lat. = 90°)
  - 1° lon. = 0 km
- Berlin (lat. = 52.513°)
  - 1° lon. = 67.9 km



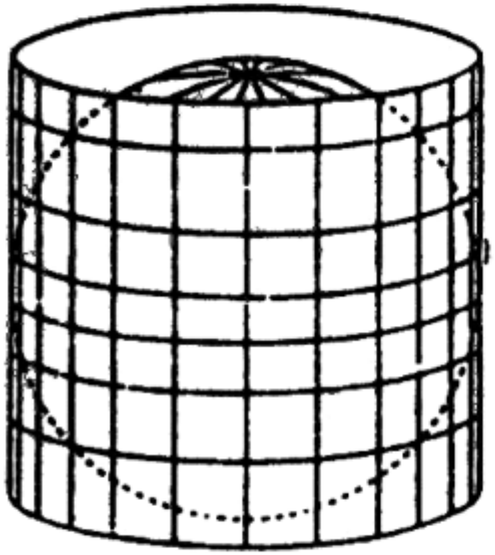


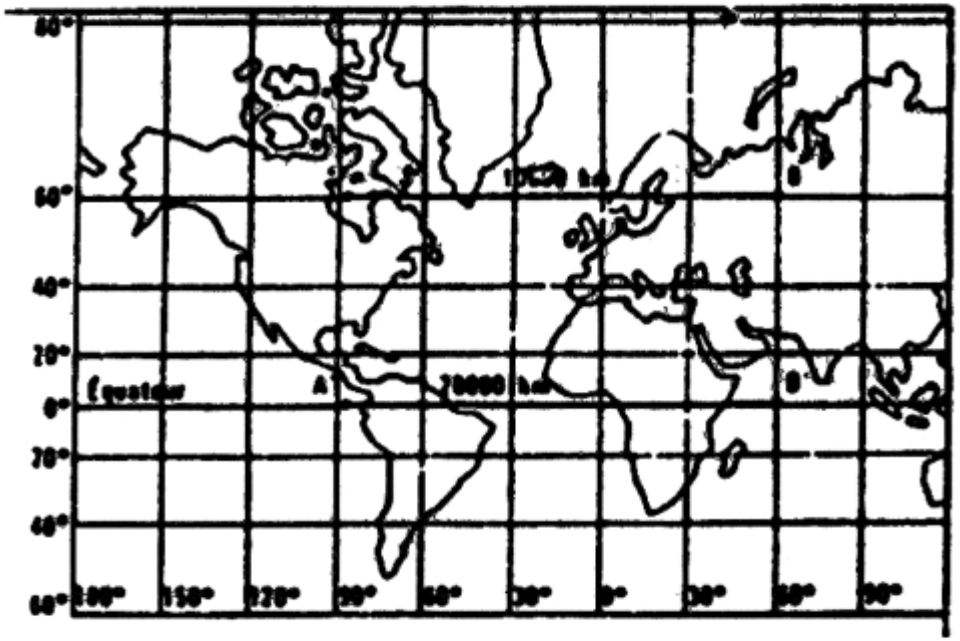


- Length of 1° of latitude slightly varies with latitude
  - Earth modeled as ellipsoid

Map Projections

- How to draw a 3D sphere on a 2D map?
- Many possibilities, all involve distortions
- Mercator projection: cylindrical projection
  - Standard for nautical navigation, used by GoogleMaps





Google Maps Zoom Levels

- Zoom level = 0
  - Equator length = 256 pixels
    = 2π * 6378137 m
  - 1 pixel = 156.5 km
- Zoom level = i
  - Equator length = 256 * 2^i pixels
    = 2π * 6378137 m
  - 1 pixel = 2π * 6378.137 / (256 * 2^i) km
- Zoom level = 19
  - 1 pixel = 2π * 6378.137 / (256 * 2^{19}) km
    ≈ 30 cm

W 9th St
5th Av
E 9th St
5th Ave

zoom level 0
equator = 256 pixels
= 2π * 6378137 m

San Lorenzo
Samanie
Túquerres
Esmeraldas
Cumbal
El Angel
Ipia
Muisne
San Gabriel
Rosa Zárate
Atuntaqui
Ibarra
Otavalo
Pedernales
Cayambe
Quito
Bahía de Caráquez
Machachi
Chone
Manta
Calceta
La Mana
Archidona
Rocafuerte
Pujilí
Tena
Quevedo
Ambato
Santa Ana
El Carmen
Balzar
Jipijapa
Guaranda
Ecuador
Puyo
Vinces

zoom level 7
equator = 32768 pixels
1 pixel = 1.2 km

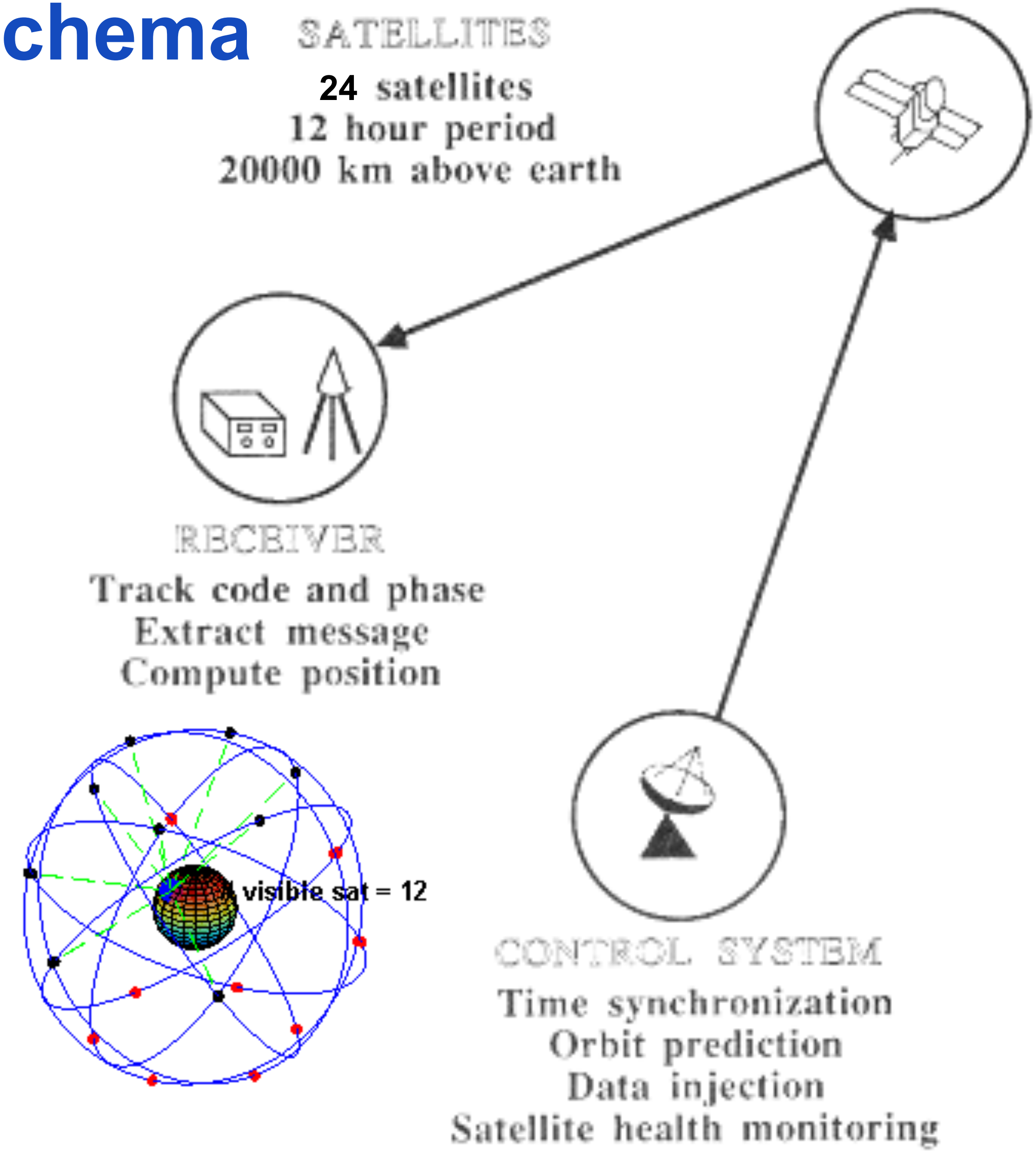
Positioning by Signal Runtime Differences

- Measuring signal runtime (“time of flight”) from known senders
- Short time spans → difficult to measure
- Problems
  - Radio: Multi-path, atmospheric distortions
  - Good placement of senders necessary
- Enhance results by introducing reference points
- Do you know a “famous” example?

GPS (Global Positioning System)

GPS: System Schema

- Components
  - Satellites
  - Control system
  - Receiver
- 24 Satellites
  - 20000 km
  - 2 times around the world a day
- Maintenance cost
  - 750M$ a year



SATELLITES

24 satellites
12 hour period
20000 km above earth

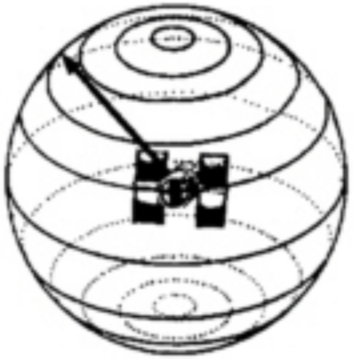
RECEIVER

Track code and phase
Extract message
Compute position

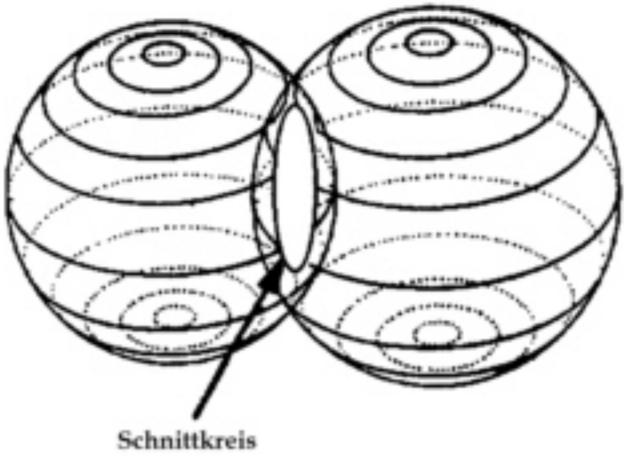
CONTROL SYSTEM

Time synchronization
Orbit prediction
Data injection
Satellite health monitoring

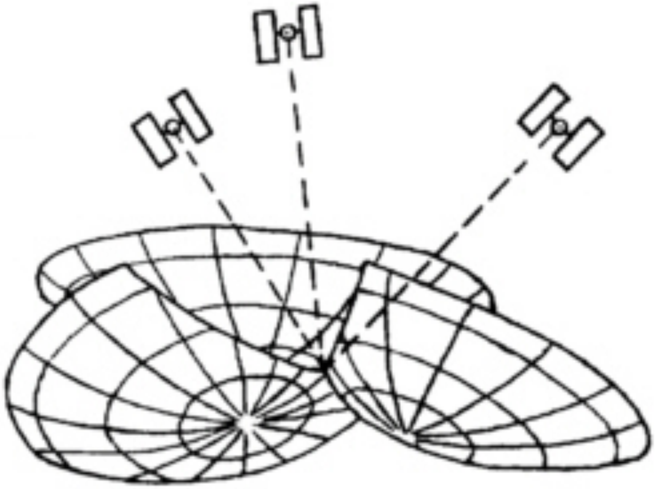
GPS: Deriving Location



one satellite



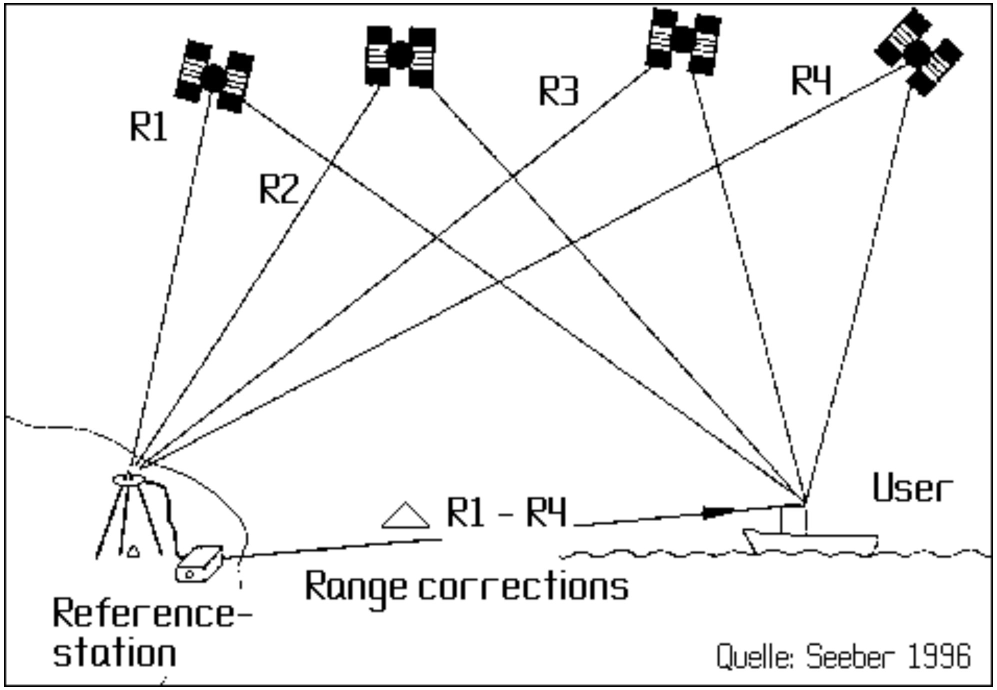
two satellites



three satellites

GPS: Differential GPS

- Enhancement of precision by using a correct reference signal (Differential GPS)
- Need to know the position of a receiver that sends the difference between actual and measured position to the mobile device
- Problem: Delay of correction signal

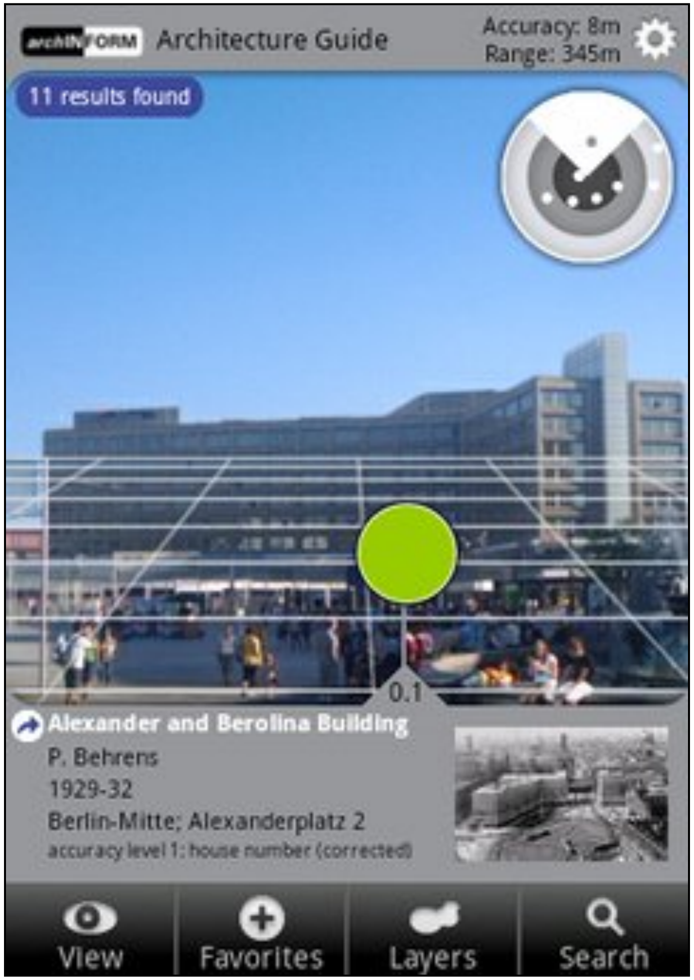


Quelle: Seeber 1996

LOCATION-BASED SERVICES AND APPLICATIONS

Layar "Reality Browser"

- Position + orientation
  GPS, accelerometer, magnetometer
- Show POIs as overlays on viewfinder image
- Platform allows inserting new layers and POIs
- Layers
  - Real estate
  - Transportation
  - Tours / Guides
  - Eating & Drinking
- http://layar.com/



Architecture Guide

Accuracy: 8m
Range: 345m

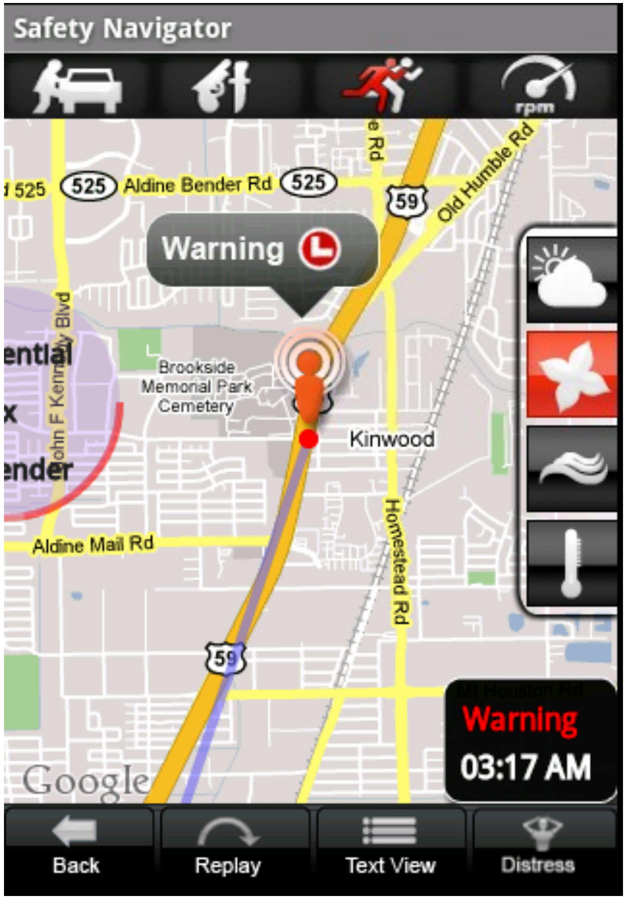
11 results found

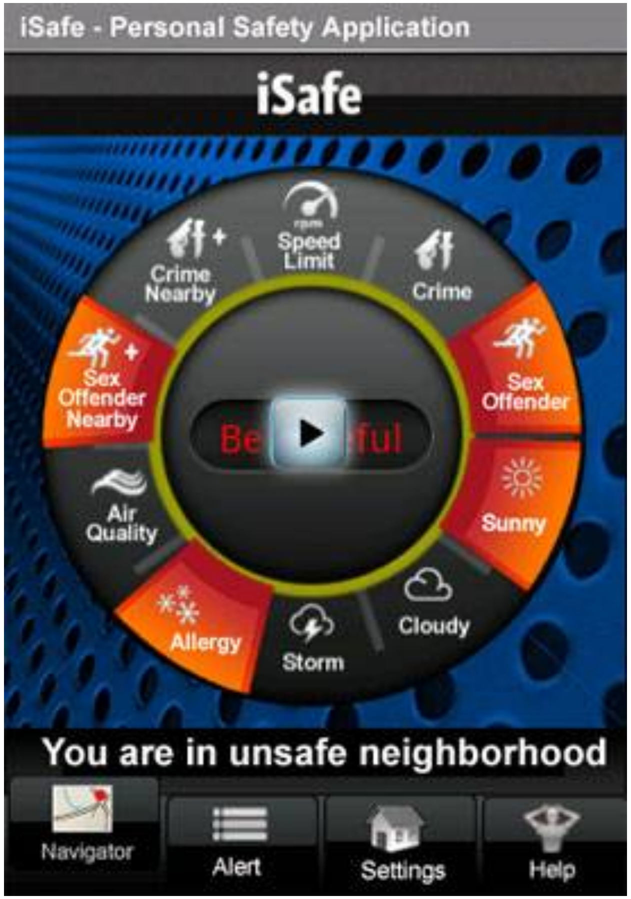
0.1

Alexander and Berolina Building
P. Behrens
1929-32
Berlin-Mitte; Alexanderplatz 2
accuracy level 1: house number (corrected)

iSafe – Personal Safety Application

- Informs you about possible dangers in your current location

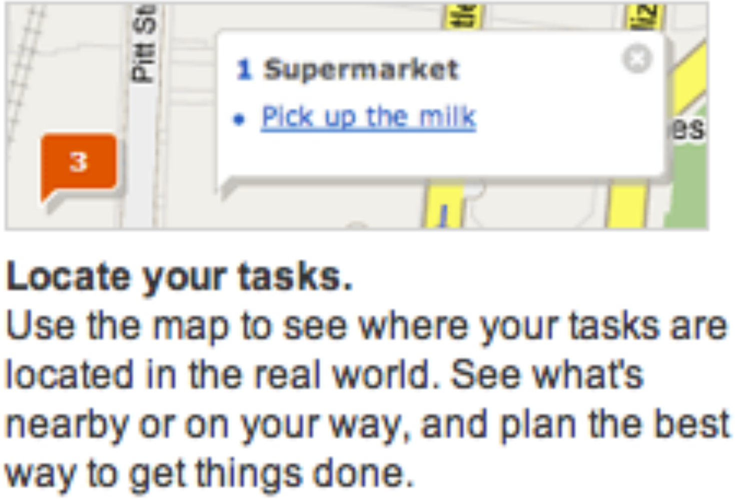




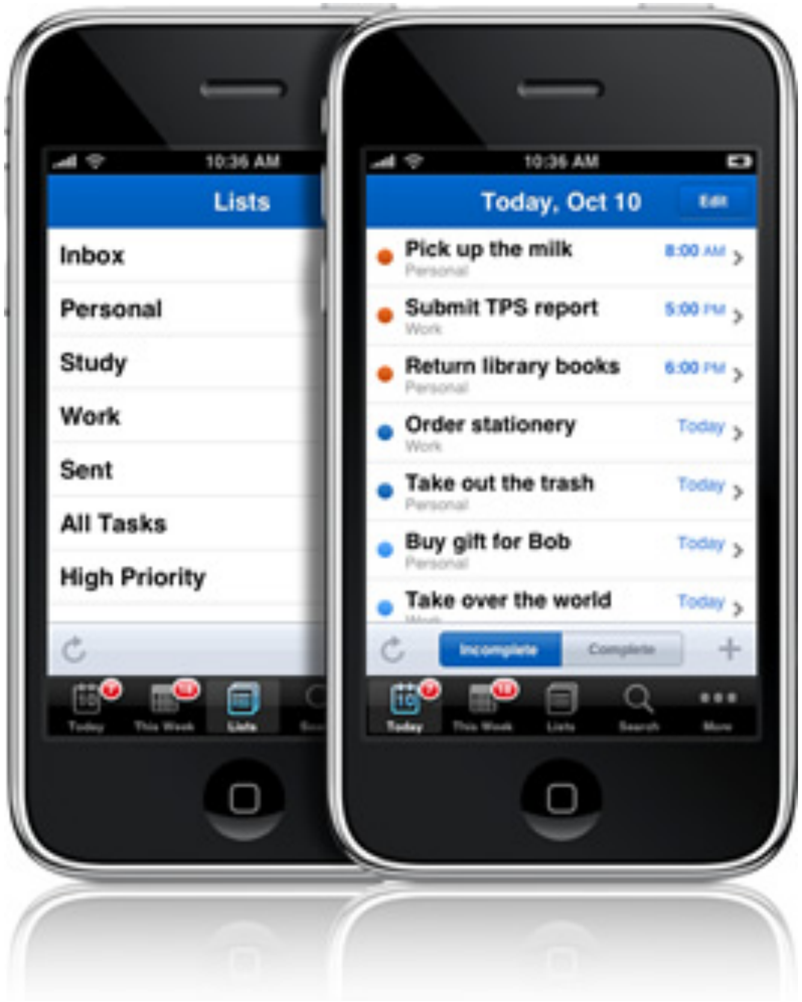
http://www.freefamilywatch.com/demo.html

Remember The Milk

- Organize your tasks by location



Locate your tasks.
Use the map to see where your tasks are located in the real world. See what's nearby or on your way, and plan the best way to get things done.



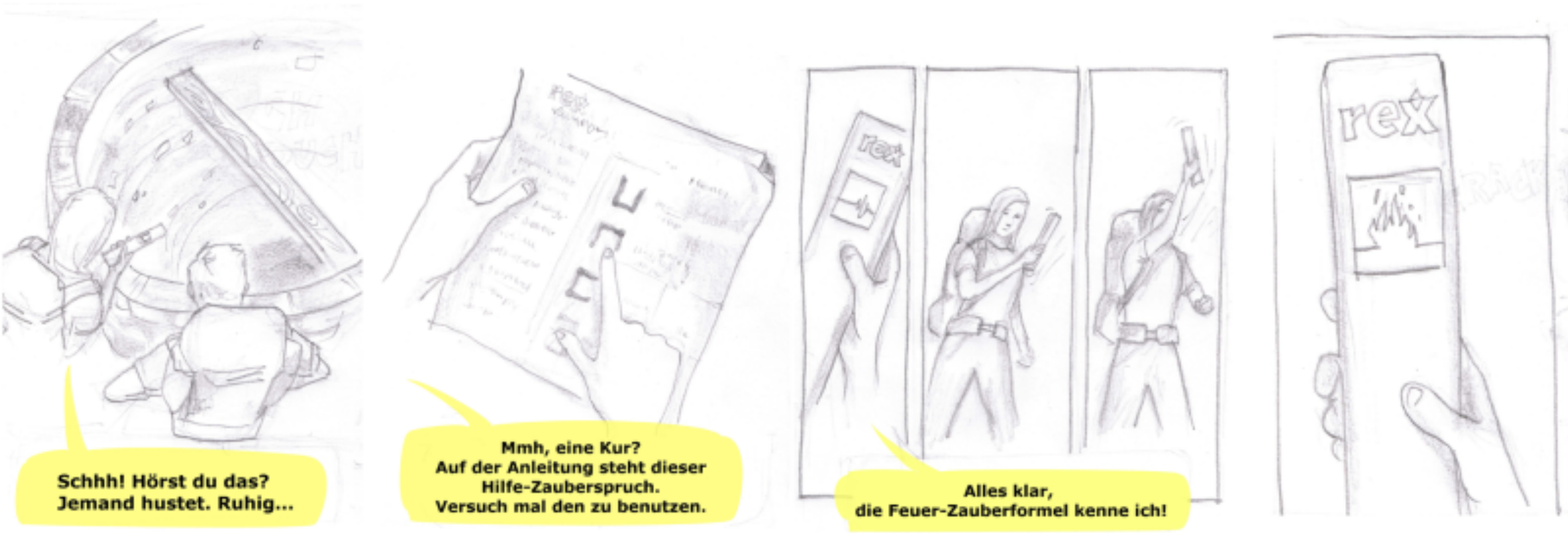
Inbox
Personal
Study
Work
Sent
All Tasks
High Priority

Pick up the milk Personal	8:00 AM
Submit TPS report Work	5:00 PM
Return library books Personal	6:00 PM
Order stationery Work	Today
Take out the trash Personal	Today
Buy gift for Bob Personal	Today
Take over the world	Today

http://www.rememberthemilk.com

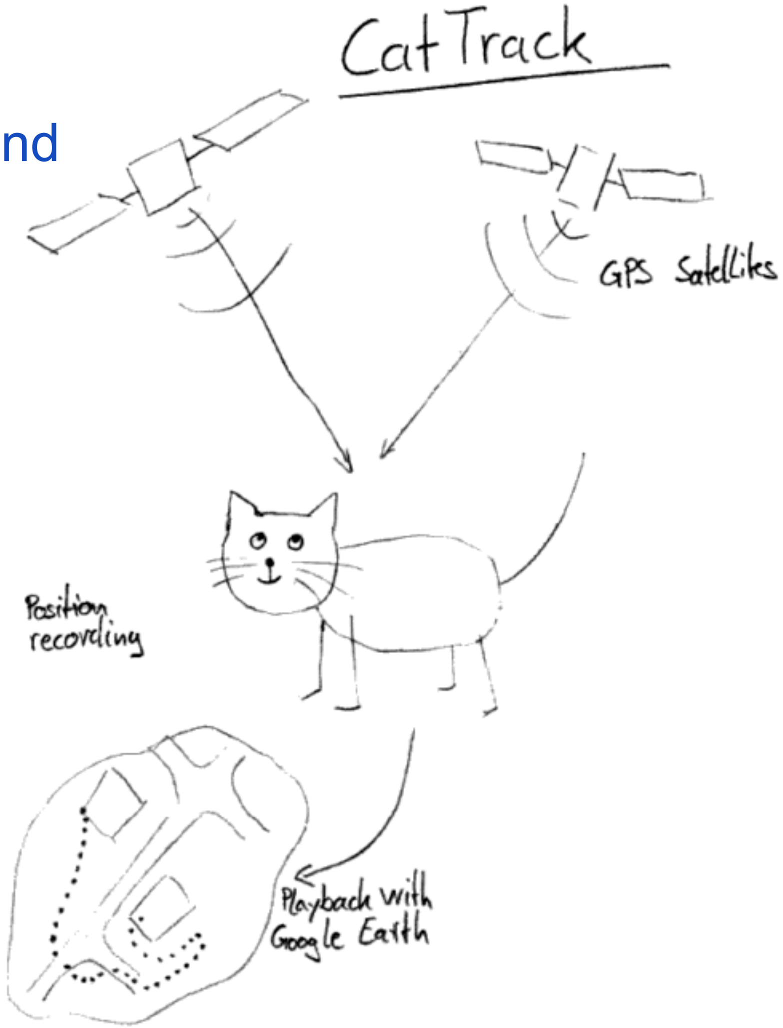
REXplorer Location-Based Game (2005+)

- Explore the ancient Regensburg and interact with historical characters through gestures
- http://www.rex-regensburg.de/stadtspiel/rexplorer
- http://hci.rwth-aachen.de/REXplorer



Mr. Lee GPS Tracking for Cats

- GPS receiver and camera
  - Know where your cat strolled around
  - Generate cat photo tours



http://www.mr-lee-catcam.de/index.htm

ANDROID LOCATION AND MAP APIS

Android Location and Map APIs

- Location API: Access location data (GPS, WiFi, GSM)
  - Package android.location
  - LocationManager
  - Geocoder

- Map API: Display and navigate maps
  - Package com.google.android.maps
  - MapView
  - MapActivity
  - Example

    developer.android.com/guide/tutorials/views/hello-mapview.html

Permissions (in AndroidManifest.xml)

- Permissions for location-based services

```
<uses-permission
  android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
  android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
  android:name="android.permission.ACCESS_MOCK_LOCATION" />
<uses-permission
  android:name="android.permission.INTERNET" />
  …
```

- Child of <application>

```
<uses-library android:name="com.google.android.maps" />
```

- Overview of Android permissions

  developer.android.com/reference/android/Manifest.permission.html

Example Manifest for Location

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="de.lmu.location"
   android:versionCode="1"
   android:versionName="1.0">
  <uses-sdk android:minSdkVersion="8" />
  <application android:icon="@drawable/icon" android:label="@string/app_name" android:debuggable="true">
    <activity android:name=".MainActivity"
         android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
  <uses-permission android:name="android.permission.VIBRATE" />
</manifest>
```

Location Manager Service

- Obtain device's geographical location
- Get notification upon entering a specified location
- Get last location
  - getLastKnownLocation(provider);
  - provider: GPS_PROVIDER, NETWORK_PROVIDER
- Register listener for location updates
  - requestLocationUpdates(provider, minTime, minDistance, listener);
  - provider: GPS_PROVIDER, NETWORK_PROVIDER
  - minTime: minimum time between notifications [ms]
  - minDistance: minimum distance between notifications [m]
  - listener: notified about location updates

Example: Last Location

```
public class LocationManagerDemoActivity extends Activity {
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    LocationManager locMgr = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);
    Location loc = locMgr
        .getLastKnownLocation(LocationManager.GPS_PROVIDER);
    Toast.makeText(this, loc.toString(), 10000).show();
    Log.d("last location", loc.toString());
    List<String> providerList = locMgr.getAllProviders();
    Iterator<String> iter = providerList.iterator();
    while (iter.hasNext()) {
      Log.d("provider", iter.next().toString());
}  }  }
```

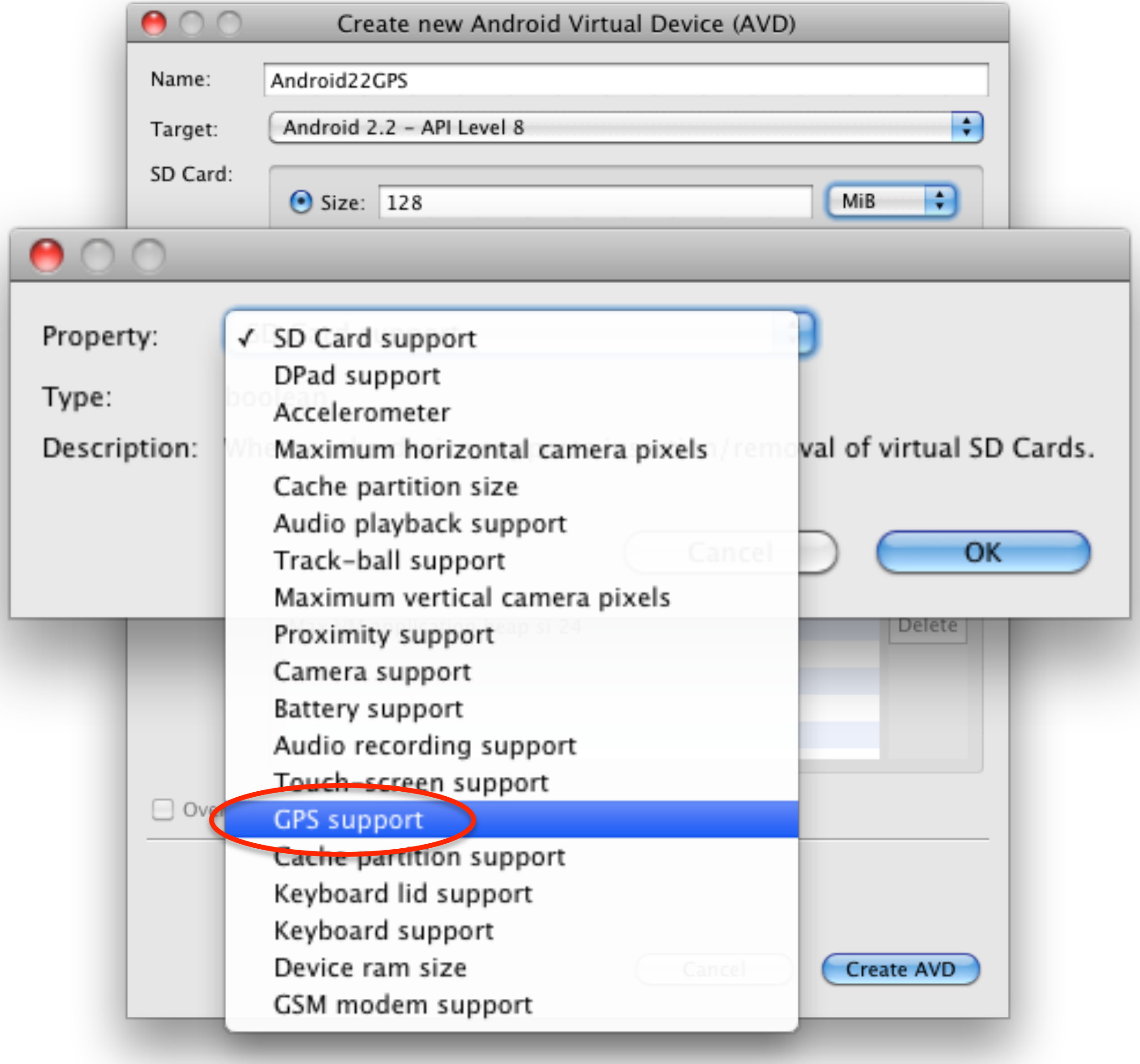
Template for Location Updates

```
public class MainActivity extends Activity implements LocationListener {
    LocationManager locationManager = null;
    …
    public void onLocationChanged(Location location) {
        if (location != null) {
            // process location update
        }
    }
    public void onProviderDisabled(String provider) {}
    public void onProviderEnabled(String provider) {}
    public void onStatusChanged(String provider, int status, Bundle ext) {}
}
```

Example: Location Updates

```
public class LocationUpdateDemoActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LocationManager locMgr = (LocationManager)
                getSystemService(Context.LOCATION_SERVICE);
        LocationListener locListener = new LocationListener() {
            public void onLocationChanged(Location location) {
                if (location != null) {
                    Toast.makeText(getBaseContext(),
                        "New location (" + location.getLatitude() + ", " +
                        location.getLongitude() + ")", Toast.LENGTH_LONG).show();
                    }
                }
                public void onProviderDisabled(String provider) {}
                public void onProviderEnabled(String provider) {}
                public void onStatusChanged(String provider, int status, Bundle extras) {}
            };
        locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER,
                5000, 0, locListener);
}}
```

Enabling GPS on the Emulator



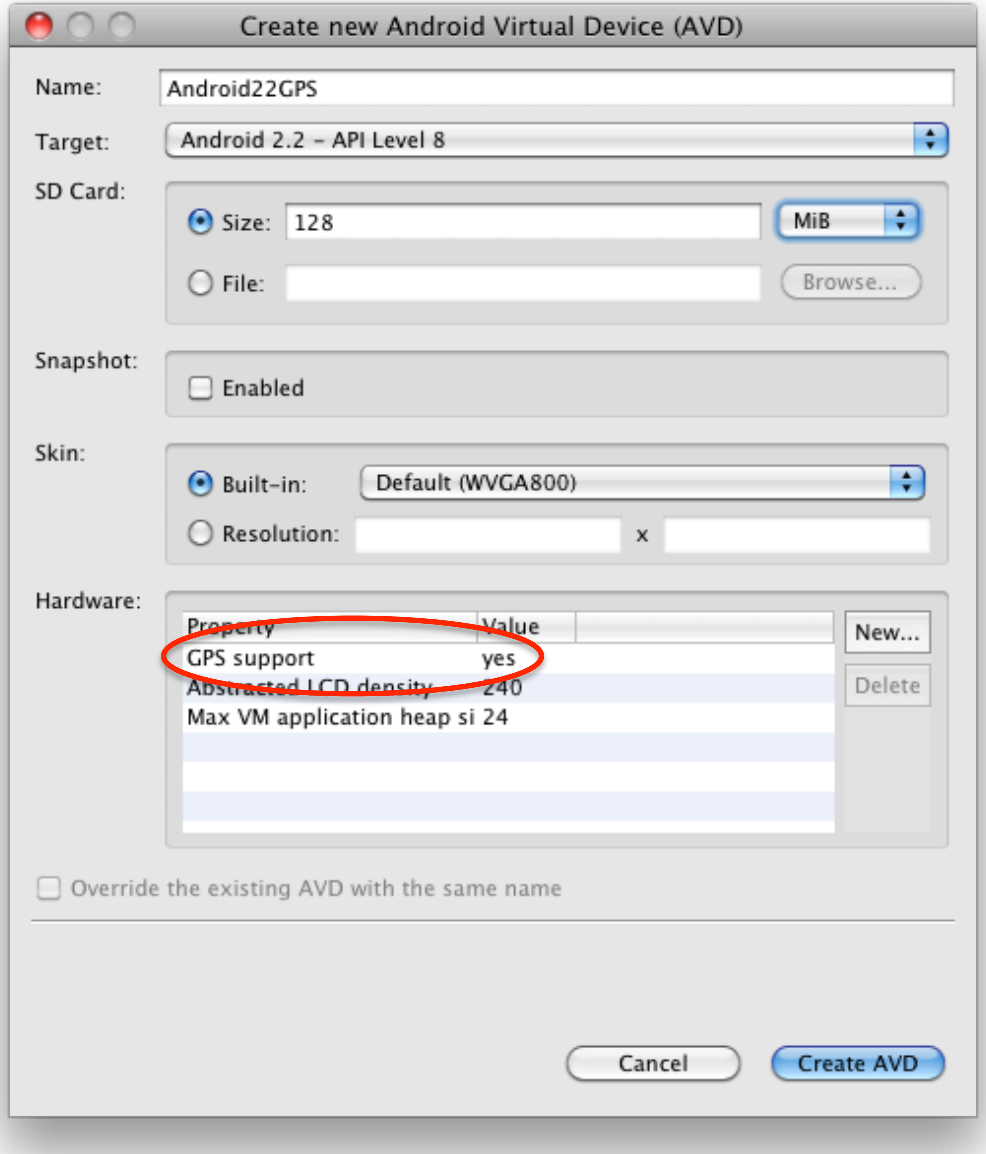
Create new Android Virtual Device (AVD)

Name: Android22GPS

Target: Android 2.2 - API Level 8

SD Card: ● Size: 128 MiB

Enabling GPS on the Emulator



Create new Android Virtual Device (AVD)

Name: Android22GPS

Target: Android 2.2 - API Level 8

SD Card:
● Size: 128 MiB
○ File:

Snapshot: □ Enabled

Skin:
● Built-in: Default (WVGA800)
○ Resolution: x

Hardware:

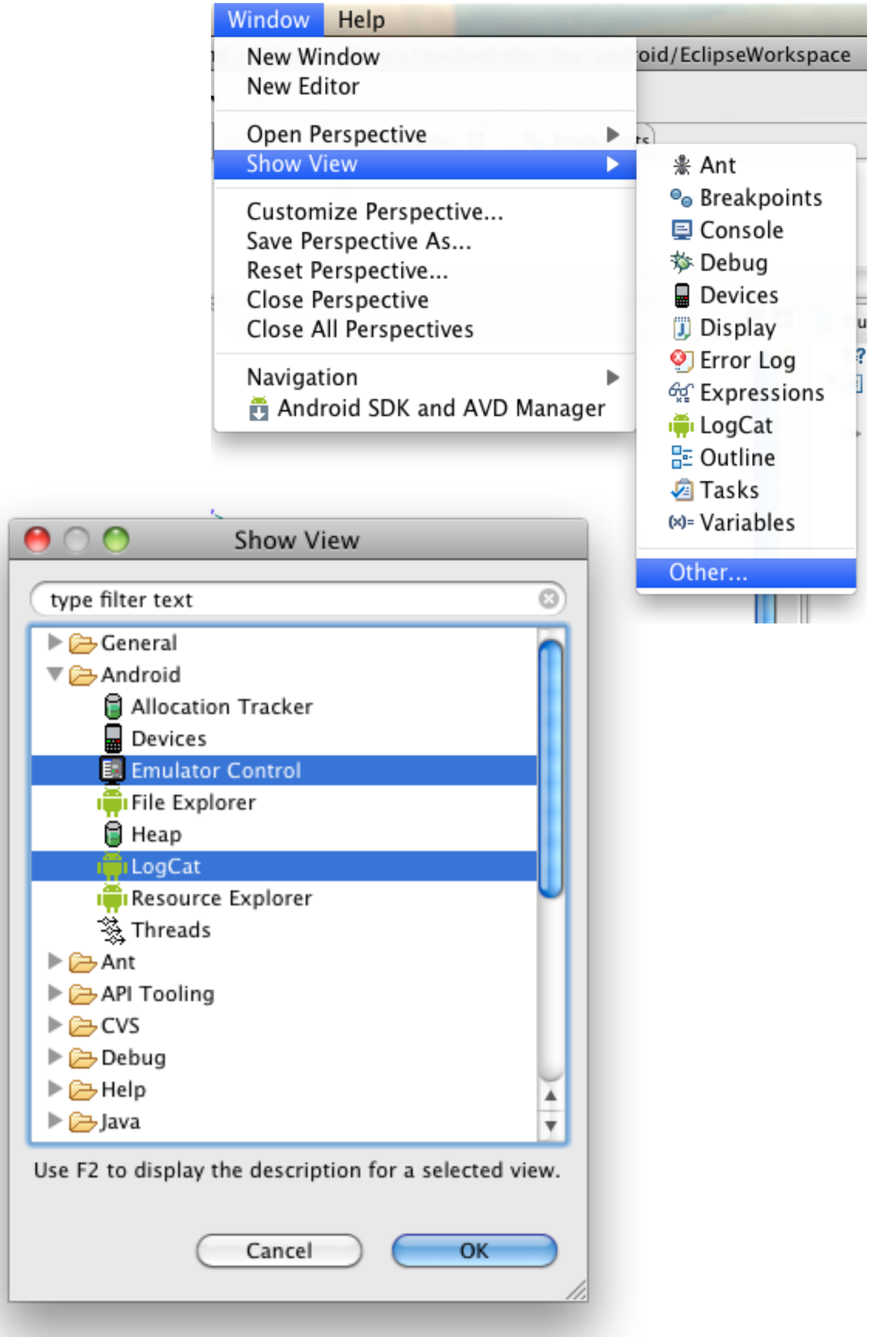
Property	Value
GPS support	yes
Abstracted LCD density	240
Max VM application heap si	24

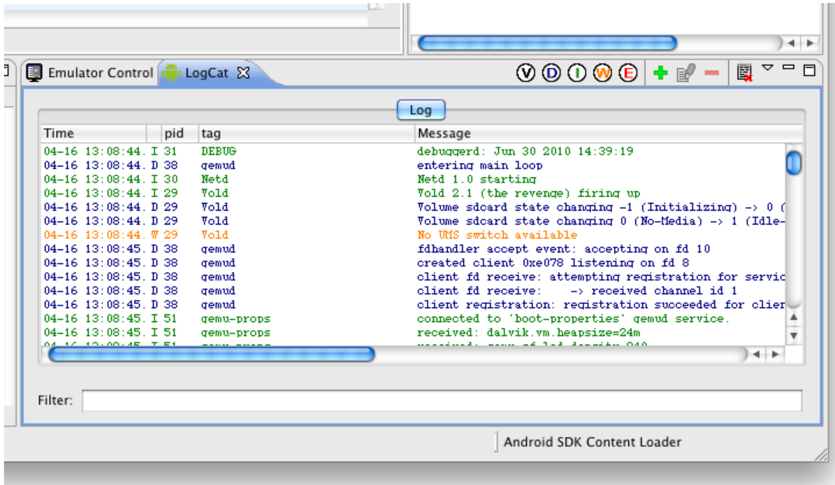
□ Override the existing AVD with the same name

Eclipse Configuration

- LogCat View
  - Log.d output

- Emulator Control View
  - Entering locations

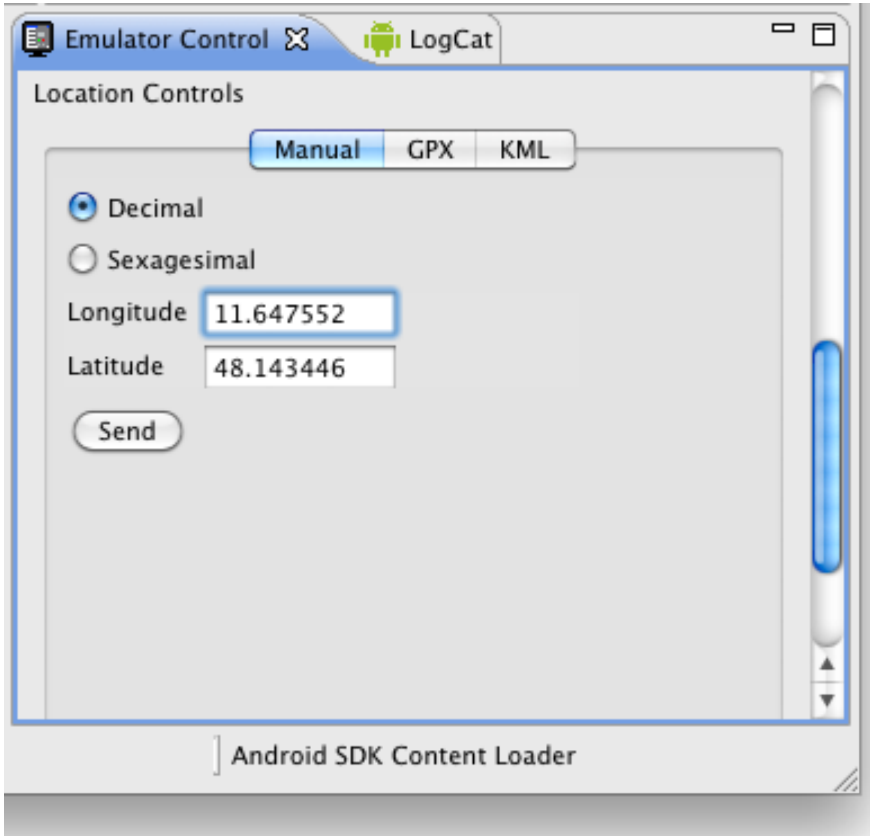




```
Time                 pid  tag          Message
04-16 13:08:44. I 31   DEBUG        debuggerd: Jun 30 2010 14:39:19
04-16 13:08:44. D 38   qemud        entering main loop
04-16 13:08:44. I 30   Netd         Netd 1.0 starting
04-16 13:08:44. I 29   Vold         Vold 2.1 (the revenge) firing up
04-16 13:08:44. D 29   Vold         Volume sdcard state changing -1 (Initializing) -> 0 (
04-16 13:08:44. D 29   Vold         Volume sdcard state changing 0 (No-Media) -> 1 (Idle-
04-16 13:08:44. W 29   Vold         No UMS switch available
04-16 13:08:45. D 38   qemud        fdhandler accept event: accepting on fd 10
04-16 13:08:45. D 38   qemud        created client 0xe078 listening on fd 8
04-16 13:08:45. D 38   qemud        client fd receive: attempting registration for servic
04-16 13:08:45. D 38   qemud        client fd receive:    -> received channel id 1
04-16 13:08:45. D 38   qemud        client registration: registration succeeded for clier
04-16 13:08:45. I 51   qemu-props   connected to 'boot-properties' qemud service.
04-16 13:08:45. I 51   qemu-props   received: dalvik.vm.heapsize=24m
```

Entering Locations in Emulator Control View

- How to get latitude and longitude?

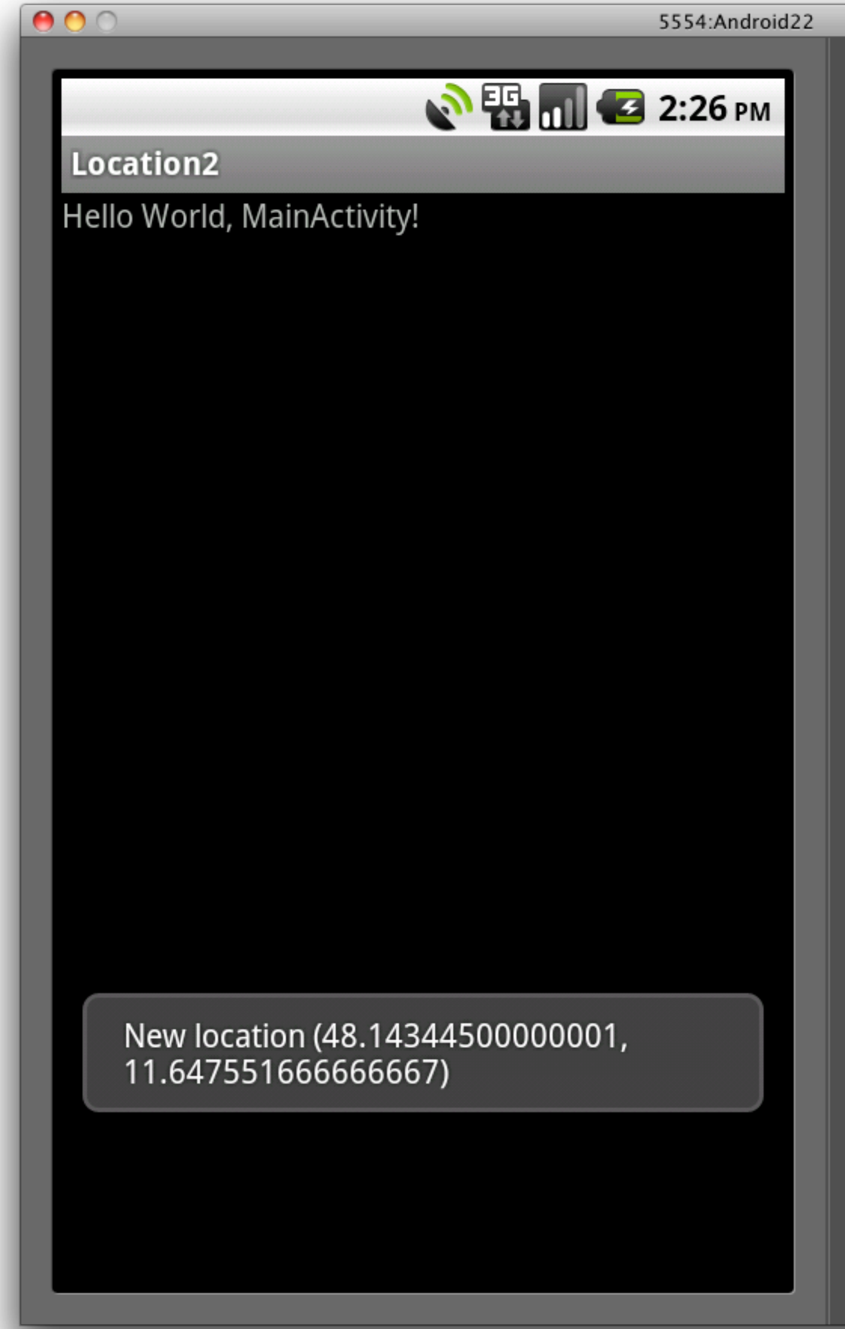


● Decimal

○ Sexagesimal

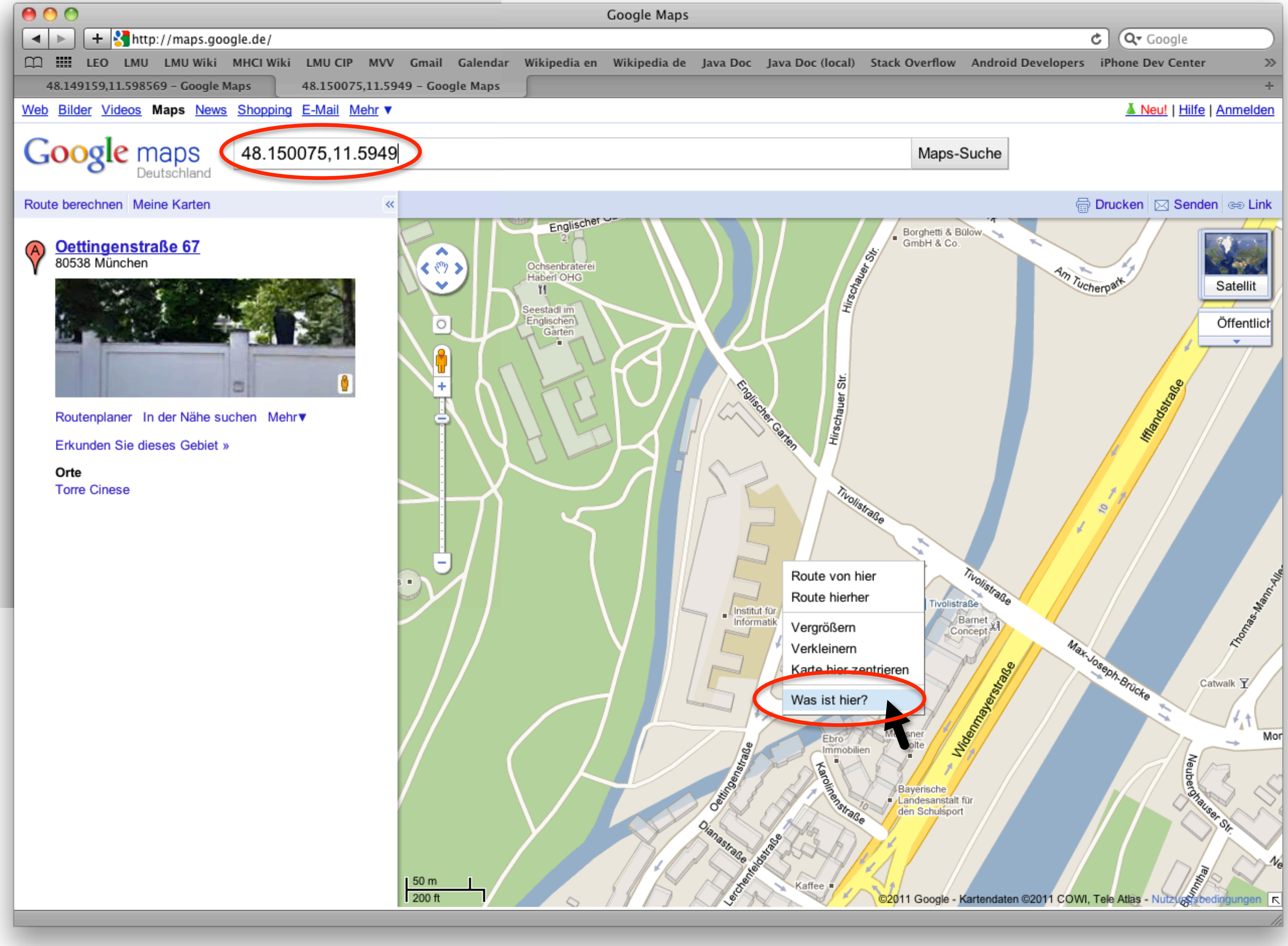
Longitude 11.647552

Latitude 48.143446



Hello World, MainActivity!

New location (48.143445000000001, 11.647551666666667)



48.150075,11.5949

Oettingenstraße 67
80538 München

Orte
Torre Cinese

Simulated Location via Telnet

- Telnet to a running emulator
  - telnet localhost <emulator port>
  - geo fix <lon> <lat>
  - geo nmea <nmea sentence>
- Example
  - telnet localhost 5554
  - geo fix 13 52
  - http://developer.android.com/intl/fr/guide/developing/tools/emulator.html

Keyhole Markup Language (KML)

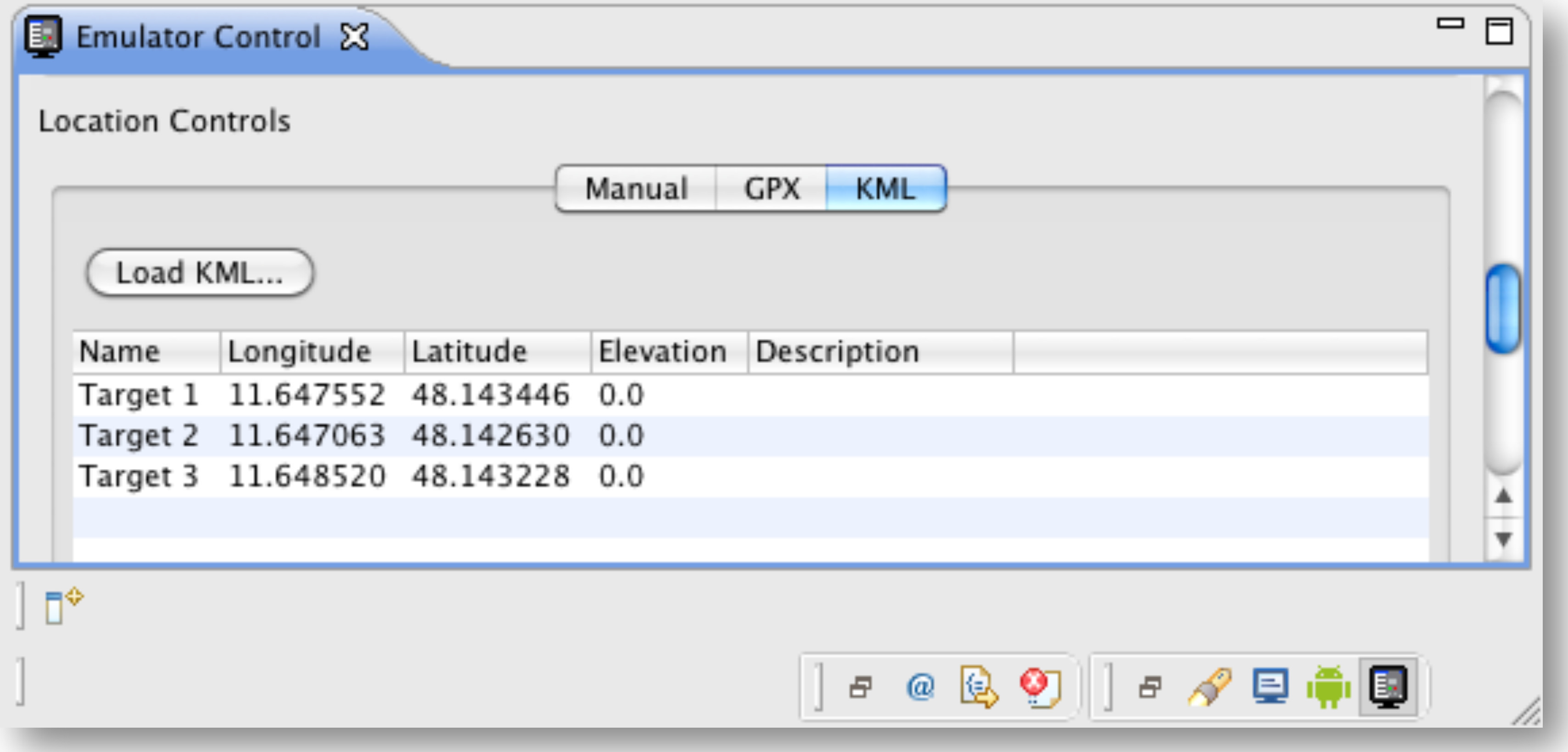
- XML-based language for expressing geographic information
  - Standardized by the Open Geospatial Consortium
  - Used in Google Maps (Mobile), Google Earth
- Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Document>
    <Placemark>
      <name>Target 1</name>
      <description>This is the first target.</description>
      <Point>
        <coordinates>11.647552,48.143446,0</coordinates> <!-- longitude, latitude, altitude -->
      </Point>
    </Placemark>
  </Document>
</kml>
```

- Try it out: kml-samples.googlecode.com/svn/trunk/interactive/index.html

KML in the Emulator

- Click a row to send location to emulator



Name	Longitude	Latitude	Elevation	Description
Target 1	11.647552	48.143446	0.0	
Target 2	11.647063	48.142630	0.0	
Target 3	11.648520	48.143228	0.0	

Distance Between Geo-Locations

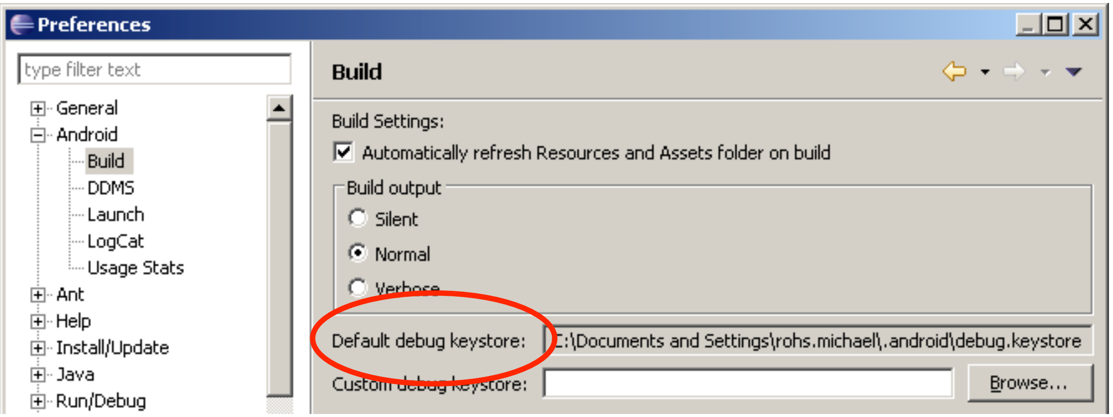
- Distance (in m) between two geolocations

```
float[] results = new float[1];
Location.distanceBetween(lat, lon, poi.latitude, poi.longitude, results);
float distance = results[0];
```

GOOGLE MAPS

Map API Key

- Locate keystore
- Open command line



Build

Build Settings:

☑ Automatically refresh Resources and Assets folder on build

Build output

○ Silent
● Normal
○ Verbose

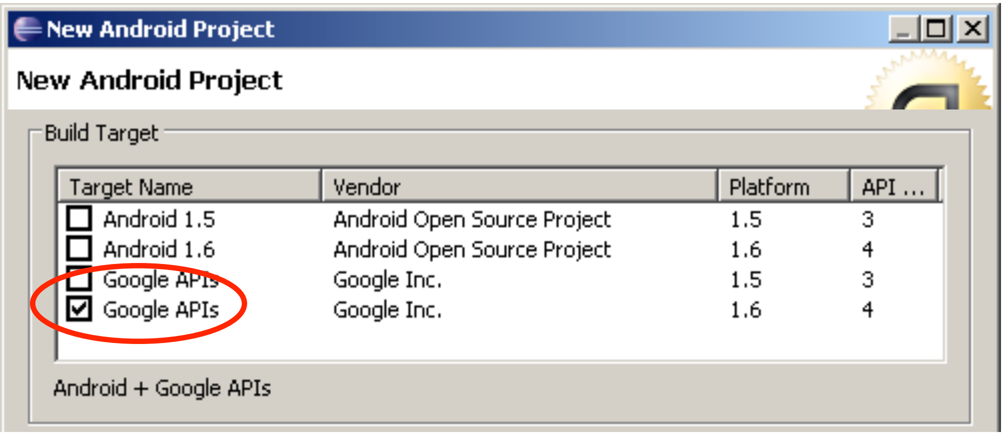
Default debug keystore: C:\Documents and Settings\rohs.michael\.android\debug.keystore

Custom debug keystore:

- Get MD5 hash of debug certificate

  keytool -list -keystore ~/.android/debug.keystore
  -storepass android -keypass android

- Get the key from Google
  - http://code.google.com/android/maps-api-signup.html
- Projects using maps need build target “GoogleAPIs”
  - Potentially needs a new AVD



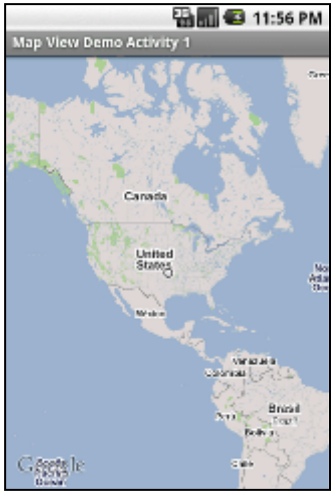
New Android Project

Build Target

Target Name	Vendor	Platform	API ...
□ Android 1.5	Android Open Source Project	1.5	3
□ Android 1.6	Android Open Source Project	1.6	4
□ Google APIs	Google Inc.	1.5	3
☑ Google APIs	Google Inc.	1.6	4

Android + Google APIs

Example Map View



- XML

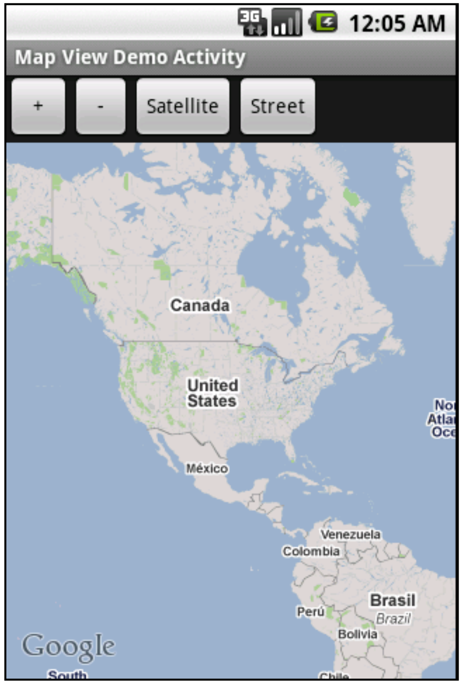
```
<LinearLayout xmlns:android="http://schemas..."
   android:orientation="vertical" android:layout_... >
   <com.google.android.maps.MapView android:layout_...
      android:apiKey="02LvHoUW1Z_HVYZWU..." />
</LinearLayout>
```

- Java

```
public class MapViewDemoActivity extends MapActivity {
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.mapview);
   }
   protected boolean isRouteDisplayed() { return false; }
}
```

Example Map View with Controls

```
<LinearLayout xmlns:android="http://schemas..."
                    android:orientation="vertical" ...>
   <LinearLayout android:orientation="horizontal" android:layout_...>
      <Button android:id="@+id/zoomin" android:text="   +   " ... />
      <Button android:id="@+id/zoomout" android:text="   -   " ... />
      ...
   </LinearLayout>
   <com.google.android.maps.MapView
      android:id="@+id/mapview“
      android:apiKey="02Lv..." ... />
</LinearLayout>
```



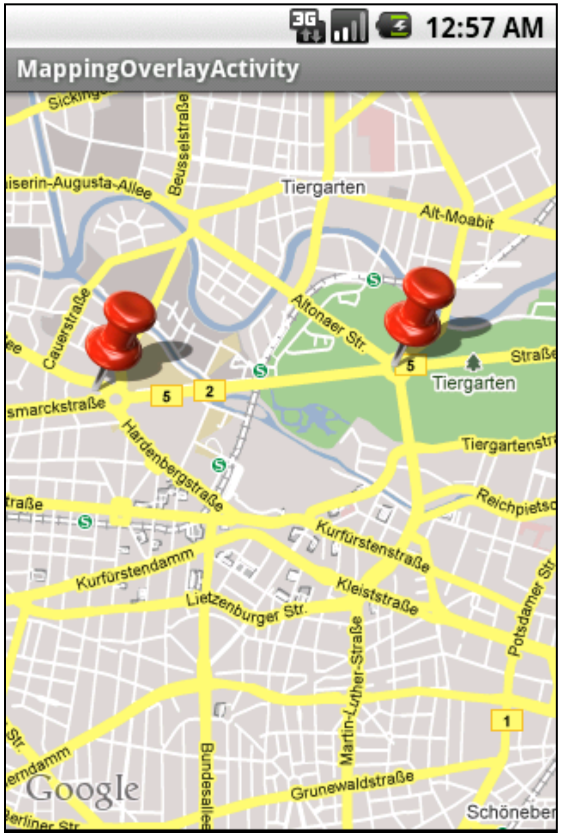
Example Map View with Controls

```
public class MapViewDemoActivity extends MapActivity {
    private MapView mapView;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapview);
        mapView = (MapView) findViewById(R.id.mapview);
        Button zoominBtn = (Button) findViewById(R.id.zoomin);
        zoominBtn.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                mapView.getController().zoomIn();
            }});
        ...
    }
    protected boolean isRouteDisplayed() { return false; }
}
```

Using Overlays

- /res/layout/mapviewoverlay.xml

```
<LinearLayout xmlns:android="http://schemas..."
                    android:orientation="vertical" ...>
    <com.google.android.maps.MapView
        android:id="@+id/mapviewoverlay"
        android:apiKey="02Lv..." ... />
</LinearLayout>
```



Using Overlays

```
public class MappingOverlayActivity extends MapActivity {
    private MapView mapView;
    private GeoPoint tlabs = new GeoPoint((int)(
        52.513036 * 1000000), (int)(13.320281 * 1000000));
    private GeoPoint saeule = new GeoPoint((int)(
        52.514495 * 1000000), (int)(13.350130 * 1000000));

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapviewoverlay);
        mapView = (MapView) findViewById(R.id.mapviewoverlay);
        mapView.setBuiltInZoomControls(true);
        mapView.setClickable(true);
        mapView.getController().setCenter(tlabs);
        mapView.getController().setZoom(14);
        Drawable marker = getResources().getDrawable(R.drawable.pushpin);
        mapView.getOverlays().add(new InterestingLocations(marker));
    }
...}
```

Using Overlays

```
class InterestingLocations extends ItemizedOverlay<OverlayItem> {
    private List<OverlayItem> locations = new ArrayList<OverlayItem>();
    private Drawable marker;
    public InterestingLocations(Drawable marker) {
        super(marker);
        this.marker = marker;
        locations.add(new OverlayItem(tlabs, "T-Labs", "T-Labs"));
        locations.add(new OverlayItem(saeule, "Siegessäule", "Siegessäule"));
        populate();
    }
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {
        super.draw(canvas, mapView, shadow);
        boundCenterBottom(marker);
    }
    protected OverlayItem createItem(int i) {
        return locations.get(i);
    }
    public int size() {
        return locations.size();
    }
}
```

Marker hotspot: bottom center

GEOCODING

Geocoding

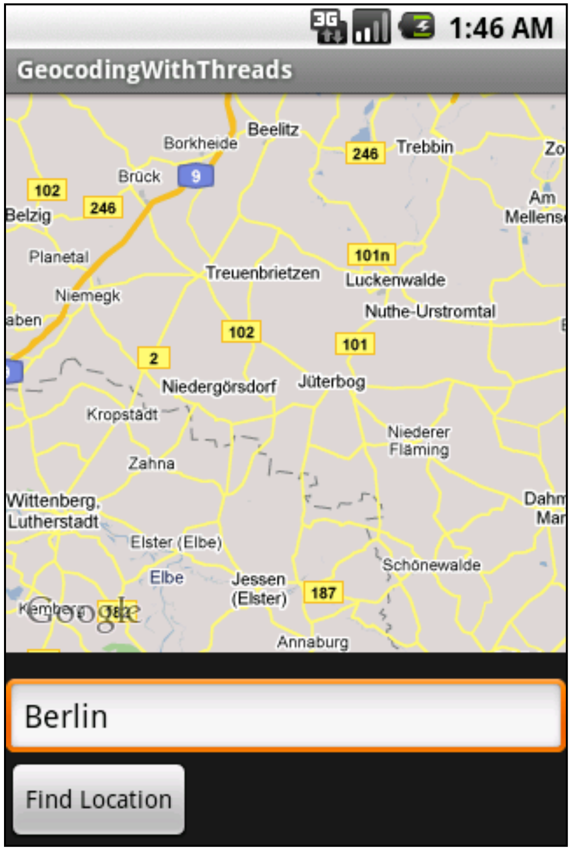
- Geocoding: address → latitude / longitude
- Reverse geocoding: latitude / longitude → address(es)

- Package android.location
  - List<Address> getFromLocation(double lat, double lon, int max)
  - List<Address> getFromLocationName(StringLocationName, int max)

Geocoding with Threads

/res/layout/geocode.xml

```
<RelativeLayout xmlns:android="http://schemas..." ...>
  <LinearLayout android:layout_alignParentBottom="true"
                android:orientation="vertical" ... >
    <EditText android:id="@+id/location" ... />
    <Button android:id="@+id/geocodeBtn"
            android:text="Find Location" ... />
  </LinearLayout>
  <com.google.android.maps.MapView
      android:id="@+id/geoMap"
      android:clickable="true"
      android:apiKey="02Lv..." ... />
</RelativeLayout>
```



Geocoding with Threads

```
public class GeocodingWithThreads extends MapActivity {
    Geocoder geocoder = null;
    MapView mapView = null;
    ProgressDialog progDialog = null;
    List<Address> addressList = null;

    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.geocode);
        mapView = (MapView) findViewById(R.id.geoMap);
        int lat = (int) (52.513036 * 1000000);
        int lng = (int) (13.320281 * 1000000);
        GeoPoint pt = new GeoPoint(lat, lng);
        mapView.getController().setZoom(10);
        mapView.getController().setCenter(pt);
```

Geocoding with Threads

```
    geocoder = new Geocoder(this);
    Button geoBtn = (Button) findViewById(R.id.geocodeBtn);
    geoBtn.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
            EditText loc = (EditText) findViewById(R.id.location);
            String locationName = loc.getText().toString();
            progDialog = ProgressDialog.show(
                    GeocodingWithThreads.this, "Processing...",
                    "Finding Location...", true, false);
            findLocation(locationName);
        }
    });
}
```

Geocoding with Threads

```
private void findLocation(final String locationName) {
    Thread thread = new Thread() {
        public void run() {
            try {
                // do background work
               addressList = geocoder.getFromLocationName(locationName, 5);
                // send message to handler to process results
                uiCallback.sendEmptyMessage(0);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    };
    thread.start();
}
```

Geocoding with Threads

```
private Handler uiCallback = new Handler() {
    public void handleMessage(Message msg) {
        progDialog.dismiss();
        if (addressList != null && addressList.size() > 0) {
            int lat = (int) addressList.get(0).getLatitude() * 1000000;
            int lng = (int) addressList.get(0).getLongitude() * 1000000;
            GeoPoint pt = new GeoPoint(lat, lng);
            mapView.getController().setCenter(pt);
        } else {
            Dialog foundNothingDlg = new AlertDialog.Builder(
                GeocodingWithThreads.this)
                   .setIcon(0).setTitle("Failed to Find Location")
                   .setPositiveButton("OK", null)
                   .setMessage("Location Not Found...")
                   .create();
            foundNothingDlg.show();
        }
}};
```

MEDIA FRAMEWORK

Media APIs

- Package android.media
  - MediaPlayer: Playing audio and video content
  - MediaRecorder: Record audio and video content
- Content sources
  - Internet
  - .apk file (as a resource or as an “asset”)
  - Secure digital (SD) card

Audio Player Example

/res/layout/main.xml

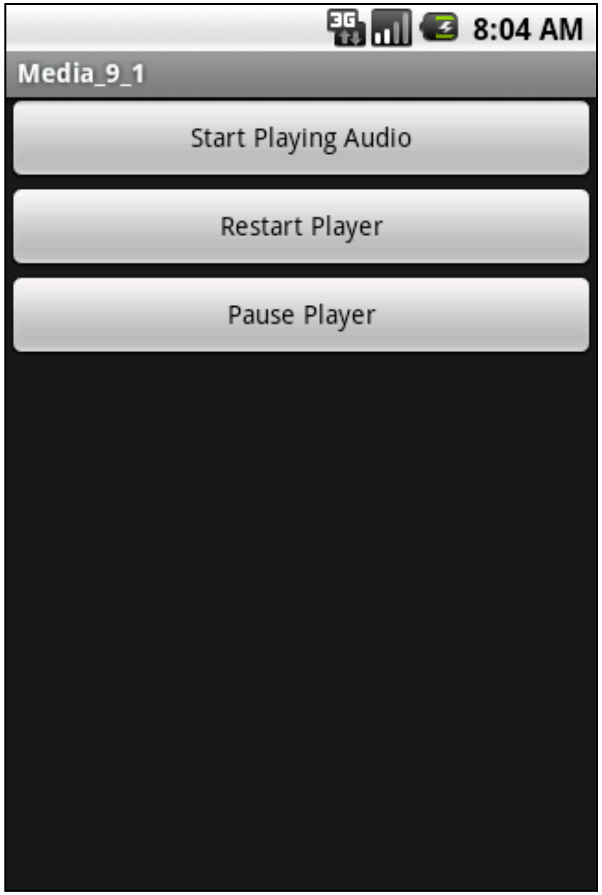
```
<LinearLayout xmlns:android="http://schemas..."
              android:orientation="vertical" ...>

<Button android:id="@+id/startPlayerBtn"
        android:text="Start Playing Audio" ... />

<Button android:id="@+id/restartPlayerBtn"
        android:text="Restart Player" ... />

<Button android:id="@+id/pausePlayerBtn"
        android:text="Pause Player" ... />

</LinearLayout>
```



Audio Player Example

```
public class MainActivity extends Activity {
    private static final String AUDIO_URL = "http://music.com/song.mp3";
    private MediaPlayer mediaPlayer = null;
    private int playbackPosition = 0;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button startPlayerBtn = (Button) findViewById(R.id.startPlayerBtn);
        Button pausePlayerBtn = (Button) findViewById(R.id.pausePlayerBtn);
        Button restartPlayerBtn = (Button) findViewById(R.id.restartPlayerBtn);
        startPlayerBtn.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                try { playAudio(AUDIO_URL); }
                catch (Exception e) { e.printStackTrace(); }
            }});
    }
```

Audio Player Example

```
    pausePlayerBtn.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
            if (mediaPlayer != null) {
                playbackPosition = mediaPlayer.getCurrentPosition();
                mediaPlayer.pause();
            }
    }});
    restartPlayerBtn.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
            if (mediaPlayer != null && !mediaPlayer.isPlaying()) {
                mediaPlayer.start();
                mediaPlayer.seekTo(playbackPosition);
            }
    }});
}
```

Audio Player Example

```
private void playAudio(String url) throws Exception {
    killMediaPlayer();
    mediaPlayer = new MediaPlayer();
    mediaPlayer.setDataSource(url);
    mediaPlayer.prepare();
    mediaPlayer.start();
}
protected void onDestroy() {
    super.onDestroy();
    killMediaPlayer();
}
private void killMediaPlayer() {
    if (mediaPlayer != null) {
        try { mediaPlayer.release(); }
        catch (Exception e) { e.printStackTrace(); }
    }
}
```

Playing Local Media Files

- In /res/raw/song.mp3

```
mediaPlayer = MediaPlayer.create(this, R.raw.song);
mediaPlayer.start();
```

- Via file descriptor

```
AssetFileDescriptor fileDesc =
        getResources().openRawResourceFd(R.raw.song);
mediaPlayer = new MediaPlayer();
mediaPlayer.setDataSource(fileDesc.getFileDescriptor(),
        fileDesc.getStartOffset(), fileDesc.getLength());
fileDesc.close();
mediaPlayer.prepare();
mediaPlayer.start();
```

The End