Location-Awareness

Petteri Nurmi
About the course

- Advanced course, algorithms and machine learning subprogramme
- 4 credit units for passing the course
  - At the end of the course, possibility to undertake a separate project for further 2 cu
- Course exam + exercises
  - Passing the course requires obtaining at least 50% of points from both
About the course

• Course webpage
  • [http://www.cs.helsinki.fi/courses/582684/2012/k/k/1](http://www.cs.helsinki.fi/courses/582684/2012/k/k/1)
  • Course material and exercises will be available in electronic form from the webpage

• Exercises
  • Exercises have to be returned electronically to the instructors, min. 15 minutes before the exercise session
  • The exercises will be available on Thursday from the course webpage
  • Solutions published online after the exercise session
Prerequisites

• Mathematics:
  • Understanding of basic concepts in geometry, linear algebra and calculus
  • Knowledge about basic concepts in probability and statistics

• Programming:
  • Capability to implement algorithms for processing location measurements
  • Implementation language can be chosen freely, but preferred languages Matlab, python, Java
Objectives

- Understand how location information can be measured and represented
- Obtain basic knowledge about approaches to positioning and be able to apply this in practice
- Learn basics of spatial data analysis and be able to identify patterns from location measurements
Outline

- Location-Awareness
  - Coordinate systems and other basic concepts
  - Location-Based Services
- Positioning and Location Systems
  - Positioning Algorithms
  - Measuring Location
- Spatial Analysis
  - Clustering, Indexing, Preprocessing, Filtering
  - Place Detection, Trajectory Mining, Recommendations
- Focus on techniques/algorithms for analyzing and processing location data
What the course does not cover?

- System level issues
  - Architectures for location-awareness
  - Middleware
- End-User aspects
  - User acceptance and experience
  - Attitudes, privacy concerns etc.
- GIS and Spatial Databases
  - Queries
  - Representation formats
Questions

- What location-awareness means and what kind of application areas are there?
- How locations on the Earth are represented?
- What is a geographic reference system?
- What is a horizontal datum and what is a vertical datum? Where are they used and how?
- How to visualize location information using Internet-based applications?
Location-Awareness

- Location-awareness refers to the capability to obtain information about the location of an entity
  - Originally the term was used to refer to configuration settings of network systems
    - Enabled applications to adapt their behavior based on available network connections
    - These days called network location awareness (NLA)
  - Nowadays focused on consumer-oriented mobile applications and services
    - Location-based services, covered in the second lectures
Examples of early systems - ParcTab

- Developed at Xeroc PARC (Palo Alto Research Center), starting 1992
  - Considered the first prototype of a ubiquitous computing system, but also one of the first location-aware systems
  - Palm-sized mobile computer that uses infrared to connect to infrastructure
- Provides information services within an office environment (PARC lab)
  - Location-sensitive: environment control (light switching), reverse paging (i.e., reporting last seen location of user)
  - Location-insensitive: calendar, thesaurus, weather, …
Examples of early systems - ParcTab

Source: http://www.ubiq.com/parctab/tabpic.html

Source: http://www.ubiq.com/parctab/tspic.html
Examples of early systems – Cyberguide

- A handheld mobile tourist guide developed at Georgia Tech (GVU Center), 1995
  - One of the first mobile guides
  - Schematic black and white maps for navigation
  - Information services about predefined indoor and outdoor areas
  - Infrared beacons used for indoor positioning
  - GPS used for outdoor positioning
Examples of early systems – Cyberguide
Examples of early systems – Guide

- PDA based tourist guide for the city of Lancaster, UK
  - Developed by Lancaster University (≈ 1999-2000)
- Position information obtained from transmitters that are positioned within the city
- Supports location-sensitive information delivery
  - Get information about the current area
  - Find more information about an object that is “visible”
- Other location-related functionalities:
  - Map navigation
  - Creating a custom tour of the city and following it
Examples of early systems – Guide

Source: http://www.guide.lancs.ac.uk/screenshots.html
Spatial Location

- Position that is based on a reference system
  - Divides the geographic area where positions are provided into smaller units
- A reference system consists of
  - Coordinate system: how positions are represented
  - Datum: model that defines how coordinates relate to the geographic area
  - Projection: mapping that determines how coordinates are visualized on maps
Geographic Coordinate Systems

• Systems for representing spatial locations as vector of numbers, called *coordinates*
  • Coordinates refer to position, typically measured either in terms of distances or angles
• Coordinate system consists of three components
  • Origin: intersection of the axes of coordinate system
  • Scale: the subdivision of axes into common units
  • Orientation: direction of axes
Geographic Coordinate Systems

Cartesian

- Coordinates specified by distance to axes
- Example: Earth Centered Earth Fixed
  - Geocenter of Earth is the origin
  - Scale is the Earth
  - Orientation of axes fixed with respect to Earth (and thus rotates with Earth)
  - Ignores curvature of earth and thus only suitable for line-of-sight / short distance calculations

Source: http://en.wikipedia.org/wiki/ECEF
Geographic Coordinate Systems
Ellipsoidal

- Geographic area is modeled as an ellipsoid
- Shape determined by a reference ellipsoid
- Origin defined by two reference planes that are orthogonal and cross at the geocenter
  - Horizontal plane corresponds to equatorial plane
  - Vertical plane includes Earth’s rotational axis and intersects North and South Poles
    - Specified separately in each reference ellipsoid
Geographic Coordinate Systems

Ellipsoidal

- Coordinates represented by angles between the reference lines and the line passing from geocenter to position
  - *Latitude* ($\Phi$): parallel to Equator, also called parallels (North/South)
  - *Longitude* ($\lambda$): perpendicular to Equator, also called meridians (West/East)
  - *Geodetic height*: the (perpendicular) distance between a position and the reference ellipsoid

Source:
http://en.wikipedia.org/wiki/Latitude
Geometric Datums

- Model that approximates the shape of the Earth
  - Defines size and shape of Earth in a coordinate system
  - Defines origin and orientation of a coordinate system
- Horizontal datum
  - Model that approximates Earth’s shape using a reference ellipsoid
- Vertical datum
  - Approximates the surface of Earth using a geoid
  - Mainly used to determine the height of a position with respect to mean sea level
Reference ellipsoid defined by a combination of equatorial radius and polar radius

- Equatorial radius $a$ defines the semi-major axis
- Polar radius $b$ defines the semi-minor axis

**Flattening $f$**

- Measure of the compression of a circle along its diameter
- $f = (a - b) / a$

Most reference ellipsoids of Earth defined using a combination of semi-major axis and inverse flattening

Source: http://en.wikipedia.org/wiki/Flattening
Horizontal Datums

- **Global datums**:  
  - Approximate the entire Earth  
  - Origin corresponds to the geocenter of the Earth

- **Local datums**:  
  - Approximate the shape of a local region  
  - Accurate for the particular region, but can deviate significantly in other regions  
  - Origin shifted from geocenter to align the ellipsoid as accurately as possible
WGS-84

- **WGS = World Geodetic System**
  - Originally defined in 1984, but revised regularly
  - Origin at the Earth’s center of mass
    - Zero longitude at IERS Reference Meridian (102.5m east of Greenwich)
- Defines both a horizontal and vertical datum
  - **Horizontal datum:**
    - Semi-major axis = 6378137.0 meters,
    - Inverse flattening $1/f = 298.257223563$
  - **Vertical datum:**
    - EGM96 (Spherical harmonics model of 360 degrees)
Vertical Datums

- Mean sea level varies due to gravitational anomalies - reference ellipsoid inaccurate for altitude
- *Geoid* is an approximation of the surface of the earth
  - Equipotential surface that coincides with mean sea level

1) True ocean level
2) Reference ellipsoid
3) Local plumb
4) Continent
5) Geoid

Vertical Datums

- Three different classes of *height* values
  - *Geoid* (1): Normal distance between reference ellipsoid and geoid
  - *Orthometric* (2): Height of a position normal to the geoid
  - *Geodedic* (3): Height of a position normal to reference ellipsoid
Geodetic Problems

- What are the coordinates of the point that is 5km NE from my current location?
  - Direct (or first) geodetic problem
  - Given a point together with an angle (azimuth) and distance, determine the coordinates of the destination
- What is the distance between two coordinate pairs?
  - Inverse (or second) geodetic problem
  - Given two points, determine the angle (azimuth) and distance between them
Geodetic Problems

- When distances small, both problems can be solved using basic geometry
- For larger distances, need to take into account the shape of the sphere
  - On ellipsoids, problem can be solved using series expansion of ellipsoidal integrals
  - Example of such expansion is Vincenty’s formula
    - Two iterative methods for solving direct and inverse geodetic problems
    - See http://en.wikipedia.org/wiki/Vincenty%27s_formulae
Map projections

- Method for representing surface of a sphere (or other 3D body) on a 2D plane
  - E.g., visualizing locations on a map (paper or mobile)
  - Spheres cannot be represented as planes without distortion (Gauss) ➔ map projections distort
- Different types of distortions:
  - Areal (equal-area): distorts (preserves) the area
  - Angular (conformal): distorts (preserves) the shape
  - Scale: proportion of distances distorted
  - Distance (equidistant): distorts (preserves) distances
  - Direction (azimuthal): distorts (preserves) angle between a point on a line and another point
Map projections

- **Planar:**
  - Ellipsoidal surface projected onto a plane that is tangential to one point
  - Distances and directions from center point accurate, but shapes and sizes distorted elsewhere

- **Conical:**
  - Flat projection surface mapped into a cone that is placed on top of the ellipsoid and unrolled

- **Cylindrical**
  - Similar to conical, but uses a cylindrical shape instead of a cone
Map Projections – Mercator

- Cylindrical map projection
  - Widely used for nautical navigation in the middle ages
  - Presented by Gerardus Mercator in 1569 and still forms the basis of many other projections
  - Conformal, i.e., preserves local angles and shapes
  - However, distorts large objects as the scale increases
    - E.g., the size of Greenland or the poles distorted
Map Projections – Transverse Mercator

- Adaptation of the Mercator projection
  - Cylinder turned 90° with respect to rotation axis
  - Projection shares a common tangent line with central meridian
  - Conformal, preserves local angles and shapes
  - Distorts shapes and areas at extreme longitudes

- Gauss-Krüger projection
  - Transverse Mercator defined in ellipsoidal form
  - Conformal with a constant scale on the central meridian
  - Thus the closer to the prime meridian, the less distorted shapes at extreme longitudes are
Map Projections – Universal Transverse Mercator

- Variant of Transverse Mercator
  - Earth divided into 60 zones
    - Each zone has width 6° of longitude
    - Zones restricted between latitudes 80°S and 84°N to avoid large distortions at poles
  - (Secant) Transverse Mercator applied separately for each zone
    - The narrow bands enable small distortion within each zone
  - Originally used for military purposes, but increasingly being used as an international standard system
Visualizing location information – Keyhole Markup Language (KML)

- During the course we will use Internet-based visualizations of spatial data
- Keyhole Markup Language (KML)
  - XML notation for geographic annotation and visualization
  - Works (e.g.) on top of Google Earth and Google Maps
  - International standard of the Open Geospatial Consortium (OGC)

http://code.google.com/apis/kml/documentation/kmlreference.html
Visualizing location information – Keyhole Markup Language (KML)

- The basic structure of KML consists of
  - XML header, always the first line
  - KML namespace declaration, always the second line
- Placemark
  - Marks a position on the Earth’s surface
    - Uses the default icon unless defined otherwise

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Goes here</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```
Point

- Specifies the position of the placemark
- Coordinates specified within a separate tag
  - Expressed in 3D (longitude, latitude, altitude) format with respect to WGS84 reference system
  - Longitude and latitude relative to WGS84 reference ellipsoid
  - Altitude relative to EGM96 geoid

```
<Point>
  <coordinates>-122.0822035425683,37.42228990140251,0
  </coordinates>
</Point>
```
Visualizing location information – Keyhole Markup Language (KML)

- The appearance of objects (points, lines, ...) can be changed using a Style definition
  - Each Style must contain an identifier
  - Objects can adopt the style using the styleUrl tag and giving a reference to the identifier of the style

- Most important elements to know
  - IconStyle: specifies how icons are drawn (e.g., color, type, size of icons)
  - LineStyle: specifies how lines / paths are visualized, (e.g., line width and color)
  - LabelStyle: specifies how text labels next to icons appear (color and size)
Visualizing location information – Keyhole Markup Language (KML)

- Paths visualized using the LineString element
- Similar to the Point element, a coordinate element is used for defining the points to connect
- Tessellate: Boolean parameter specifying whether the line string follows the geoid (= 1) or not
- Altitude mode:
  - clampToGround: ignore altitude
  - relativeToGround: altitude relative to ground level
  - absolute: altitude relative to geoid (sea level)
- Altitude offset: allows moving each point on the line segment up/down without modifying coordinates
- Extrude: Boolean parameter determining whether line string is connected to ground

```xml
<LineString id="ID">
  <gx:altitudeOffset>0</gx:altitudeOffset>
  <extrude>0</extrude>
  <tessellate>0</tessellate>
  <altitudeMode>…</altitudeMode>
  <coordinates>
    -122.377830,37.830445,0
    -122.377576,37.830631,0
    -122.377840,37.830642,0
    -122.377830,37.830445,0
  </coordinates>
</LineString>
```
Visualizing location information – Keyhole Markup Language (KML)
Visualizing location information – Keyhole Markup Language (KML)

Untessellated line string vs. Tessellated line string

Line strings using absolute and relative extrude values
Location-awareness refers to the capability to obtain information about the location of an entity.

- First systems were developed during the 90s.
- Most were designed for indoor use, positioning based on visibility of beacons.

Position with regards to a reference system called a spatial location.

Reference system consists of:
- Coordinate system
- Datum
- (Map) projection
Summary

- Two main types of coordinate systems:
  - Cartesian: Earth centered
  - Ellipsoidal: Earth modeled as an ellipsoid
    - Latitude: parallel to Equator
    - Longitude: perpendicular to Equator
- Horizontal and vertical datums
  - Horizontal: reference ellipsoid (semi-major axis and inverse flattening)
  - Vertical: geoid, approximates mean sea level
Summary

• Map projections:
  • Planar or azimuthal
  • Conical
  • Cylindrical
  • All projections distort, UTM most accurate for international maps

• Keyhole Markup Language KML
  • XML format for annotating and visualizing locations
  • Useful for location-aware applications
Literature