Geographic Information Systems: Theory and Application

Tim Schoenharl

Notre Dame CSE

April 11, 2007

Overview

Motivation

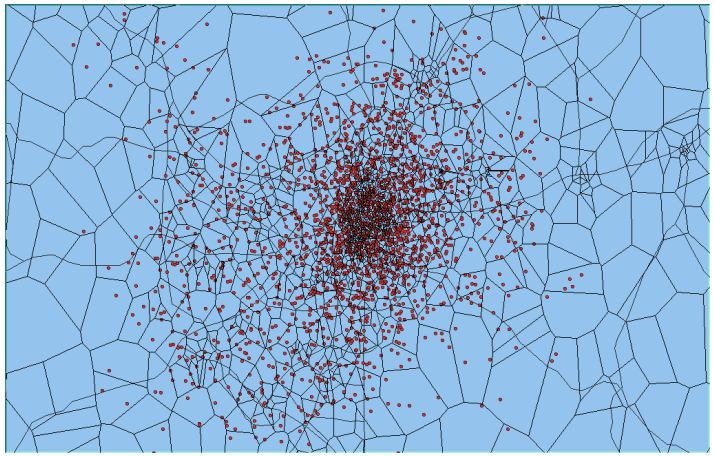

Figure: GIS Image from a WIPER Simulation

GIS Uses:

- Visualizing tower locations, relationship to urban areas, etc
- Analyzing call patterns, activity in relationship to income levels, etc
- Simulations: agent and tower locations initialized from data, agents can interact with real world geography

Sphere, Ellipsoid, Geoid

Representations of the Surface of the Earth

Sphere Simple solid, uniform radius.

Ellipsoid Solid of rotation formed from a two-dimensional ellipse. Also referred to as spheroid.

Geoid "The surface of the Earth's gravitational field".

1

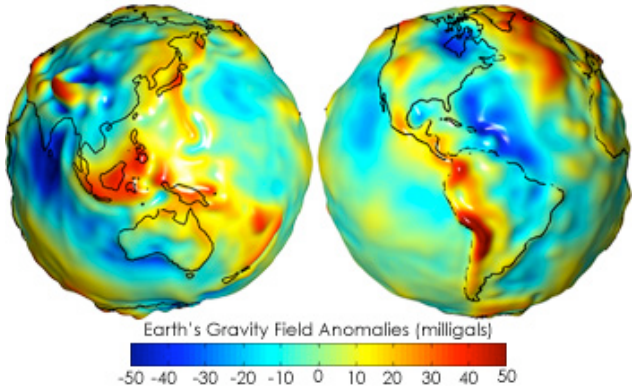

Figure: 3D plot of Geoid undulations.

[1] Image taken from `http://en.wikipedia.org/wiki/Image:Geoids_sm.jpg`

Datum

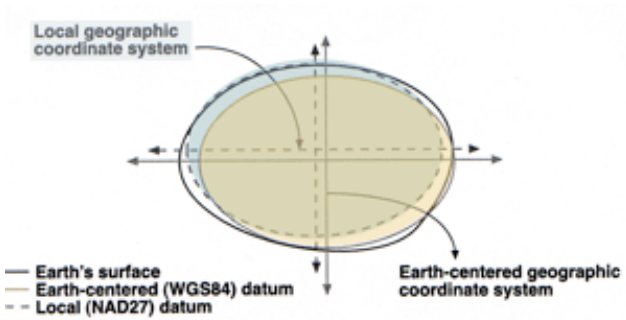

Figure: WGS 84 and NAD 27 datums compared to the shape of the Earth.

- A Datum is a representation of the surface of the Earth, an ellipsoid that has local corrections to account for the irregular nature of the Earth's surface.
- Datum are important because they relate local measurements (I am here) to a global reference (Here = lat, long (37.0625, -95.677068)).
- The most common datum and most accepted global datum is WGS 84. When you take a measurement with a GPS unit, that coordinate is likely in the WGS 84 datum.

2

[2] Figure taken from `http://www.esri.com/news/arcuser/0401/datum.html`

Projections

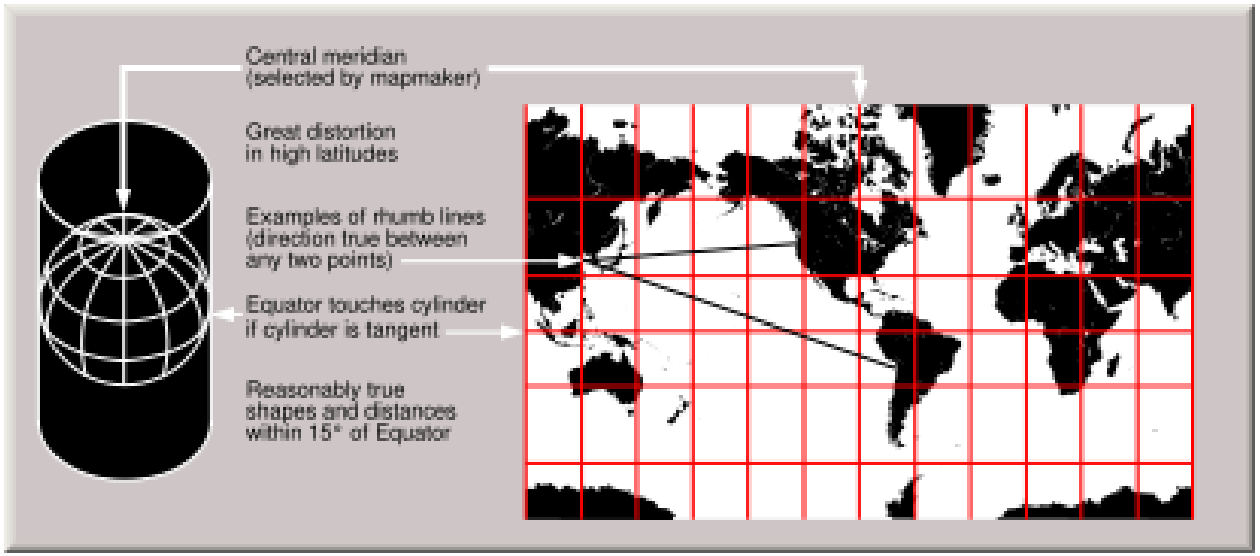

Figure: The Mercator Projection[3]

In order to visualize geographical data, either on a monitor or on paper, the data must be transformed from its spherical/geoidal reference to a 2D plane. This is the act of projecting.

[3] Image courtesy `http://en.wikipedia.org/wiki/Image:Usgs_map_mercator.svg`

Software

Software Titles

- Proprietary: ESRI ArcMap, ArcGIS, Google Earth (almost GIS)
- Open Source: GRASS GIS, OpenMap, Quantum GIS
- Open Source Libraries/Tools: GDAL, OGR, PROJ4, GEOS, JTS, GeoTools

GIS and Relational Databases

PostGIS

- http://postgis.refractions.net/
- Allows data to be stored in db tables
- Stores points, lines, polygons, etc in WKT (Well-Known Text) or WKB (Well-Known Binary) format
- Can be used as a data store for GIS systems
- Most importantly, allows users to conduct spatial analysis with SQL queries
- When using PostGIS, it is important to understand the SRID: Spatial Referencing System Identifier

SRID

srid	auth_name	auth_srid	srtext	proj4text
4326	EPSG	4326	GEOGCS["WGS 84", DATUM["WGS_1984", SPHEROID["WGS 84", 6378137, 298.257223563, AUTHORITY["EPSG", "7030"]], AUTHORITY["EPSG", "6326"]], PRIMEM["Greenwich", 0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.01745329251994328, AUTHORITY["EPSG", "9122"]], AUTHORITY["EPSG", "4326"]]	+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Table: SRID table entry

- SRID is referenced by an integer, [2000, 32766]
- SRID contains information on the agency/group that defined it and the characteristics (datum, projection, ellipsoid)
- PostGIS uses PROJ4 to convert data between different SRIDs
- PostGIS recognizes 2671 different SRIDS

SQL and Spatial Analysis

The advantages of using SQL for spatial analysis are enormous. Consider the following example: In order to determine the quality of the tower information, we wanted to quantify the error in the tower locations.

- Tower location information contains a tower ID, street address, postal code, and latitude/longitude coordinates
- In a separate file we have polygons representing the shapes of all of the postal codes
- We want to find all the towers that are "outside" their given postal code

SQL and Spatial Analysis

This can be accomplished with a one-line SQL query, joining the tables on the postal code:

```
SELECT towernum, postal_code, id FROM tower_data INNER JOIN postal_codes
ON tower_data.postal_code = postal_codes.id WHERE NOT
contains(postal_codes.the_geom_srid_4269, tower_data.the_geom );
```

We can improve this analysis by getting a list of the towers, sorted by increasing distance from the postal code, to allow for discrepancies in coordinates and datum.

```
SELECT towernum, postal_code, id, distance(tower_data.the_geom,
postal_codes.the_geom_srid_4269) FROM tower_data INNER JOIN postal_codes
ON tower_data.postal_code = postal_codes.id WHERE NOT
contains(postal_codes.the_geom_srid_4269, tower_data.the_geom ) ORDER BY
distance(tower_data.the_geom, postal_codes.the_geom_srid_4269)
```

SQL and Spatial Analysis

Gotchas

Be ware of several issues when using PostGIS for spatial analysis:

- All data must be in the same SRID for spatial queries to be valid
- Data that is loaded using shp2pgsql without an SRID will come in with SRID = -1
- Spatial queries will return results in DEGREES. For results in meters, you will need to first reproject the coordinates into an SRID that has a datum in meters, then convert your query to meters.
- When choosing an SRID, best choices are ones with +ellps=WGS84 and +units=m.
- Find these by querying the spatial_ref_sys table and searching in the "proj4text" string

Working with GRASS

- When starting GRASS, it will ask for a bounding box around the region you are viewing.
- GIS files come in two types, raster and vector. These define the commands you will use for data manipulation in GRASS.
  - For vector files: use v.in, v.out
  - For raster files: use r.in, r.out
  - Convert from vector to raster using v.to.rast
- Build voronoi cells by importing the list of points, then calling v.voronoi
- Associate data with the cells using v.to.rast input=inputvector output=outputraster use=attr column=attribute
- Images can then be exported

Sample Images from the WIPER Project

Figure: Cell phone activity overlaid on a satellite image.

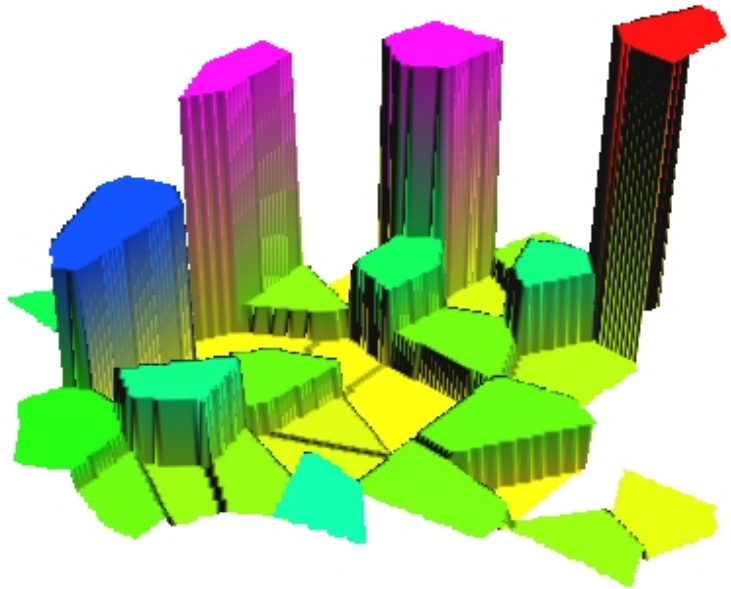

Figure: 3D View of Tower Activity

References

For More Information:

- Lavender Wiki GIS page: `http://lavender.cse.nd.edu/mywiki/index.php?title=GIS`
- PostGIS Manual: `http://postgis.refractions.net/docs/`
- Grass Homepage: `http://grass.itc.it/`
- OpenMap: `http://openmap.bbn.com/`
- GeoTools: `http://geotools.codehaus.org/`

Questions