Lecture III
TIGER DATA SET

There is usually a gap between the real world data formats and the data structures which are theoretically the most appropriate for algorithms. Witness the plethora of graphics file formats, and the many trade books that treats them. In this lecture, we will describe an important format called TIGER (Topologically Integrated Geographic Encoding and Referencing System) developed by the U.S. Bureau of the Census (http://www.census.gov). For more information, we recommend going to the TIGER homepage: http://www.census.gov/geo/www/tiger.

§1. Overview

The TIGER data set contains the geographic encoding of the whole USA, that is especially appropriate for representing census information, its primary application. For our purposes, we are mainly interested in its map data.

TIGER files. The data set is county-based, so that each county in the county has its own data set. The whole USA has over 3200 counties. Each county is given a 6 digit identifier called an FIPS. For instance, the FIPS for New York County (i.e., Manhattan) is 36061 while Kings County (i.e., Brooklyn) is 36047. The first two digits of the FIPS identifies the state (New York State is 36).

Each county has up to 17 different record types, and all the records of a given type for the county is stored in a single file. So, there are up to 17 files for each county. The files for Manhattan are named TGR36061.RT1, TGR36061.RT2, TGR36061.RTA, etc. The suffix RT1, RT2, etc, tells us that the file contains "Record Type 1", "Record Type 2", etc. The file suffixes are RTn where n is one of the characters

$$1, 2, \ldots, 9, A, C, H, I, P, R, S, Z.$$

Each file is an ASCII file. Each record is stored in one line of the file, and each line has a fixed number of ASCII characters. Each record also has a fixed number of fields, and fields occupy predetermined positions in its line.

Coordinate System and Accuracy. The coordinate system is based on Longitude and Latitudes. Each Longitude is given as a signed 9 digit sequence, with an implied 6 decimal places. Thus -123456789 really represents -123.456789. Each latitude is given as a signed 8 digit sequence, also with an implied 6 decimal places. Thus +12345678 really represents +123.45678. For instance, the bounding box for New York State is (minLon, maxLon, minLat, maxLat) = (-79.762418, -71.778137, 40.477408, 45.010840). Note that a negative Longitude means West of Greenwich Meridian). Also, "longitude 123456789" does not mean 12 degrees, 34', 56.89", etc.

Accuracy. The accuracy of the map is that of a 1:100,000 scale map, which is correct up to ± 167 feet. However, the relative positions of any specified points (with respect to the plane subdivision of the Tiger data) is correct.

Version. The latest version of the TIGER data is 2000. TIGER line information across different versions are not necessarily consistent.

TIGER Geometry. There are three kinds geometric objects: points, polygonal lines and polygons. The polygonal lines in TIGER files have the registered name of TIGER/Line (R). They are also known as "complete chains", but in this lecture, we simply call them **Tiger Lines**. Each Tiger Line has a unique ID (or TLID). Each polygon, which we will call **Tiger Polygons**, also has an ID (or POLYID), but this ID is unique only within each county. But is easy to make a POLYID unique across the whole country by concantenating it with the FIPS, for instance. Each Tiger Line is basically the maximal polygonal chain that share the same "line features". We will discuss line features below, but the type of a line feature might be "road" or "county boundary". The basic Tiger Line information is stored in RT1 and RT2.

Points (or vertices) in a Tiger line are of two kinds: of Tiger Lines, and non-endpoints. The latter are called **detail points**. Every Tiger Line has two endpoints, and zero or more detail points. In the Tiger File RT1, the endpoints are stored with each Tiger Line. The detail points are stored in RT2. Since Tiger files have fixed size records, while the number of detail points for a Tiger Line has no a priori bound, the solution is to group the detail points for a Tiger Line into groups of ten detail points per record in RT2 files. At most one group has less than 10 detail points. If a Tiger Line has no detail points, then it has no corresponding entry in the RT2 file.

It is important to note that points in the Tiger Line do not have independent existence (i.e., no unique ID's). This could potentially lead to inconsistencies, both within a county and across two adjacent counties. Inconsistencies cannot happen with detail points, as they are not shared with other Tiger Lines. But endpoints will generally be shared by two or more Tiger Lines. If two Tiger Lines L, L' share an endpoint P, then the coordinates of P in L and in L' must agree. This consistency is not encoded into the Tiger data organization, but is an implicit guarantee.

The set of Tiger Polygons forms a subdivision of its county (and by extension to the whole USA). Each Tiger Polygon is associated with an interior point. The boundary of each Tiger Polygon is bounded by a whole number of Tiger Lines; this implies that each Tiger Line bounds one or two Tiger Polygons. A Tiger Lines bounds only one Tiger Polygon if the other side of the line is outside the county or outside the map coverage.

Non-Geometric and Non-topological Data. Such data include census data (of course), postal address ranges, zip code, land type and metropolitan areas. Also of practical interest is landmarks (school, park, airport, etc).

The table below lists some important fields in the various Tiger files:

Record Type	Field Name	Field Position	Description
RT 1	TLID	6-15	TIGER/Line ID
	SIDE1	16	Single side? (blank = both sides, 1 = single side)
	FRLONG	191-200	Start (from) longtitude
	FRLAT	201-209	Start (from) latitude
(Basic info	TOLONG	210-219	End (to) longtitude
for Lines)	TOLAT	220-228	End (to) latitude
RT 2	TLID	16-25	TIGER/Line ID (not all lines have an entry here
	RTSQ	16-18	Record Sequence Number (1, 2, etc)
	LONG1	19-28	Start (from) longtitude of point 1
	LAT1	29-37	Start (from) latitude of point 1
(Detail points	LONG2	28-37	Start (from) longtitude of point 2
for Lines)	LAT2	etc	etc
RT A (Basic info for Polygons)	POLYID	6-15	Polygon ID
RT I	TLID	6-15	TIGER/Line ID
(Line-Polygon	POLYIDL	27-36	Polygon ID on the left side
info)	POLYIDR	42-51	Polygon ID on the right side
RT P	POLYID	16-25	Polygon ID
(More info	POLYLONG	26-35	Internal Point Longtitude
on Polygons)	POLYLAT	36-44	Internal Point Latitude

§2. Issues

The main conversion is to extract from the Tiger dataset the usually connectivity expected in structures such as half-edge data structures. What makes this interesting is that we may want to process this incrementally (such as across the netwok).

Distance: we often need to evaluate the distance between two points in earth (Lon/Lat) coordinates. The following information is taken from the paper "TSPLIB95" by Gerhard Reinelt, University of Heidelberg, URL http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSTLIB.html. Suppose the coordinates x, y of point (x, y) are in the form $DDD°MM'$, but represented as a decimal as $DDD.MM$. Assume the earth is a perfect sphere of radius $R = 6378.388$ kilometers. We convert $x = DDD.MM$ to radians using the function $rad(x)$

Pi = 3.141592;
deg = (int)x;
minute = x - deg;
rad(x) = PI * (deg + 5.0 * minute / 3.0) /180.0;

Then the distance $D(p, p')$ between $p = (x, y)$ and $p' = (x', y')$ is given by

q1 = cos(rad(x) - rad(x'));
q2 = cos(rad(y) - rad(y'));
q3 = cos(rad(y) + rad(y'));
D(p,p') = (int) (R * arccos(0.5*(1.0 + q1)* q2 - (1.0 - q1)*q3)) + 1.0);

When we display maps based a global referencing system (usually this means the longitude/latitude system), it is seldom acceptable to treat this as if the coordinates come from a rectangular coordinate system. Otherwise, the map would look distorted. We need to choose some map projection scheme. That is, if G is the surface of the spherical globe, we want to choose a projection surface P and a partial map $\pi : G \to P$. The simplest projection is the **sterographic projection**. In this case, P is a plane tangent to G at any chosen point $p_0 \in G$. For any point $p \in G$, we define $\pi(p) \in S$ to be the intersection of S with the ray $\rho(p)$ that emanates from the center of the globe and passing through p. Note that $\pi(p)$ is undefined for points in a hemisphere and $\pi(p_0) = p_0$.

Another simple projection is to take P to be cylinder that touches G at the equator. Again, the projection point $\pi(p)$ is defined to the intersection of $\rho(p)$ with P. In this case, $\pi(p)$ is undefined for only two points, the North and South poles. This **cylindrical projection** is is easy to calculate if $p \in G$ is given in terms of longitude-latitude $(\theta, \phi) \in [-\pi, \pi) \times (-\pi/2, \pi/2)$. and the cylinder P has the rectangular coordinates $[-\pi, \pi) \times \mathbb{R}$. Assume that points (θ, ϕ) on the equator projects to $\pi(\theta, \phi) = (\theta, 0)$. We leave this as an exercise.

When we merge counties, there may be inconsistencies or unexpected topology. For instance, we discovered in merging Manhattan with neighboring New Jersey counties that the Liberty Island belongs to Manhattan County but is surrounded by water belonging to New Jersey. How do we check for topological consistency and regularity?

Exercises

Exercise 2.1: Write a Java program to pull out all Tiger Lines and Polygons that lie within some longitude/latitute range. ♢

Exercise 2.2: Derive formulae for the projections discussed above:
(a) stereographic projection with respect to a point p_0,
(b) cylindrical projection. ♢

Exercise 2.3: (a) Write a program which, given a computer file directory (storing all the Tiger files for a single county), will extract the minimum bounding box (in terms o longitude/latitude) for the county.
(b) Construct a data base for the entire USA storing a minimum bounding box for each county.
(c) Using this, write a program to construct a list of all the counties that (potentially) intersects a given longitude/latitute range. ♢

Exercise 2.4: Write a program for checking the topological consistency of the data in (a) a single Tiger county, and (b) in several Tiger counties. Basically we need to check that the endpoints of Tiger Lines are consistent. ♢

Exercise 2.5: Write a program which, given a directory storing the Tiger files for a single county, and a postal address, will compute and determine the longitude/latitude of a point near this address. Extra credit: also extract the zip code. ♢

THE END