

# **DEVELOPING AN APPLICATION FOR MEASURING ACCURATE FEATURE DISTANCES AT REGIONAL AND GLOBAL SCALES**

**Jeong Chang Seong, PhD**

Associate Professor

Department of Geosciences, University of West Georgia

1601 Maple Street, Carrollton, GA 30118

[jseong@westga.edu](mailto:jseong@westga.edu)

678-839-4051 (Tel.); 678-839-4071 (Fax)

and

**Jinmu Choi, PhD**

Assistant Professor

Department of Geosciences, Mississippi State University

108 Hilbun Hall

Mississippi State, MS 39762

[jc778@msstate.edu](mailto:jc778@msstate.edu)

662-325-5803 (Tel.); 662-325-9423 (Fax)

## **ABSTRACT**

Calculating accurate distances with a dataset that covers a large area is not a simple task. Current geospatial software packages mostly calculate them on Cartesian map projections. With a practical need to calculate feature distances at regional and global scales, this research project developed an application that allows users to calculate geodesic distances with vector shapefiles. A software class module was developed for accurate geodesic calculations. The application supports 55 pre-defined ellipsoids and users can also define their own ellipsoids. The application was tested with the U.S. Interstate Highways and the Chattahoochee and Apalachicola river basin. Results showed that the application is very effective and convenient in calculating accurate distances at regional and global scales.

Keywords: geodesic distance, ellipsoidal distance, shapefile

## **INTRODUCTION**

The distance calculation with geographic information systems (GIS) datasets is not accurate at regional and global scale when the Pythagorean Theorem is used with the Euclidean coordinates that are referenced to a geographic map projection. Even though computer-implemented GIS is known to be very accurate, most GIS software packages calculate geographic distances based on a map projection, which naturally inherits errors caused by the uneven scale factors in map projections. For example, the ellipsoidal distance between the La Guardia airport (73.877 ° W, 40.774 ° N) in New York and the L.A. International airport (118.409 ° W, 33.942 ° N) in Los Angeles is 3 973 796 meters. However, it changes to 3 954 741 meters with the Lambert conformal conic projection (central meridian of 96 ° W, standard parallels of 33 ° N and 45 ° N, and latitude origin of 39 ° N), and 3 942 280 meters with the Albers conic projection (central meridian of 96 ° W, standard parallels of 29.5 ° N and 45.5 ° N, and latitude origin of 23 ° N). Furthermore, when a more complex dataset, such as interstate highways or rivers in the United States, is used, distance calculation errors are systematically affected by the map projection that is used for referencing X and Y coordinates.

There are limited tools that calculate the distances of line features with a GIS dataset when the features spread over a large geographic area. Examples include the GeoMedia™ (Intergraph, 2003) and the Geodetic Toolkits from the National Geodetic Survey (NGS – <http://www.ngs.noaa.gov>). These software packages use the ASCII text file or the NGS Bluebook format file as input, which are not familiar in the GIS field.

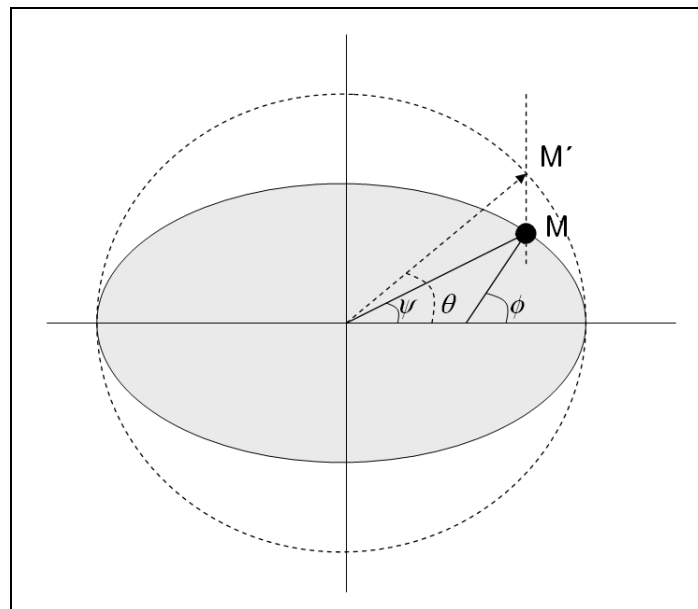
In this study, a C++ program (i.e. GEODIST) was developed to allow geodesic distance calculation with a popular GIS data format. GEODIST reads the Shapefile format (ESRI, 1998) as input and uses the comma separated

value (CSV) file format as output. With GEODIST, a user can calculate ellipsoidal distance, great circle distance and approximate distance on regional-scale or global-scale GIS datasets.

## GEODESIC DISTANCE WITH EARTH SHAPES

The Earth's shape has been defined in many different ways – sphere, prolate ellipsoid, oblate ellipsoid, pear, and geoid (Defense Mapping School, 1973), which affects distance measurement significantly. The highest accuracy in distance measurement can be obtained with the potato-like Earth shape. However, the complexity of calculation increases exponentially as the study area increases because of the irregular geoidal surface.

In GEODIST, the ellipsoidal definition of the Earth was used because geographic latitudes and longitudes are using oblate ellipsoids of revolution. There are many latitudes, but only three are of practical importance (Ewing and Mitchell, 1970). These are the geographic latitude (a.k.a. geodetic latitude), the geocentric latitude, and the reduced latitude. At a point M on an ellipsoid, the geographic latitude ( $\phi$ ) is the angle between the normal of a point on an ellipsoidal surface and the equatorial plane (Figure 1). The geocentric latitude ( $\psi$ ) is the angle between the equatorial plane and the line that connects the center of ellipsoid and the point M. The reduced latitude ( $\theta$ ) is the angle between the equatorial plane and the line that connects the center of ellipsoid and the new point M'. M' is the intersection of the major circle of the ellipse of which radius is the semi major axis and the extended line from the point M along the y axis. GEODIST uses the geographic latitude as input. The geocentric and reduced latitudes should be converted to the geographic latitude with the equations of  $\phi = \text{atan}(a^2/b^2 \cdot \tan(\psi))$  and  $\phi = \text{atan}(a/b \cdot \tan(\theta))$ , where  $a$  is the semi major axis and  $b$  is the semi minor axis (Ewing and Mitchell, 1970).



**Figure1.** Various Latitudes

A geodesic is the shortest path between two points on the surface. In the plane the geodesics are straight lines, while they are great circles on the sphere. On the ellipsoid of revolution, the geodesic line has a double curvature and results in 'S-shape pattern' between two normal sections. Geodesics should not be confused with normal sections. The normals between two points on the ellipsoidal surface of revolution never intersect because of flattening. The normal section is a curved line which connects two points and lies in a plane which contains the normal at the one point and passes through the other point.

For an ellipsoidal surface given parametrically by  $x = x(u, v)$ ,  $y = y(u, v)$ , and  $z = z(u, v)$ , the geodesic can be found by minimizing the arc length of  $\int \sqrt{dx^2 + dy^2 + dz^2}$ . Because  $dx$ ,  $dy$  and  $dz$  are defined by partial differentials, geodesics can be calculated by integrating partial differentials. There are many solutions that calculate the geodesic distance between two points on an ellipsoid. Bomford (1971) lists eight formulae in his book – Cunningham’s azimuth formula, Rudoe’s formula, Robbins’s formula, USC & GS formula, Helmert’s formula, Hyperbolic formula, Sharma’s formula, and Topographical formula. Vincenty (1975) shows compact formulae for the direct and inverse solutions of geodesics of any length, from a few centimeters to nearly 20 000 km. Using the Rainsford (1955)’s inverse formula, Vincenty developed an iterative solution. Hooijberg (1997) developed a BASIC program that calculates geodesic distance between two points using Kivioja’s method. This study adopted the Vincenty’s formula, which is used by NGS, to develop a C++ classes.

## ARCHITECTURE OF GEODIST

GEODIST is composed of three modules – CGeodesic class for distance calculation, shapelib.dll functions (Warmerdam, 1999) for shapefile handling, and CGeoDist class for user interface. The CGeodesic class has the following methods and member variables:

```
class CGeodesic {
private:
    double    a;                // semi major axis
    double    b;                // semi minor axis
    double    c;                // polar radius of curvature
    double    f;                // flattening
    double    e2;               // square of eccentricity.
    double    ep2;              // square of second eccentricity of ellipse
    double    pi;               // defined as acos(-1.0).
    double    E0, E2, E4, E6, E8; // used for direct calculation of meridional arc

public:
    struct EllipsoidType
    {
        int    ID;
        char    Name[40];
        double SemiMajorAxis;
        double InverseFlattening;
    } Ellipsoids[56]; // Ellipsoid DB
    void    PopulateEllipsoids(); // This fills in the Ellipsoid DB. Constructor calls this.
    CGeodesic(); // constructor
    virtual ~CGeodesic(); // Destructor
    void    SetEllipsoid_ab(double semiMajorAxis, double semiMinorAxis);
    void    SetEllipsoid_af(double semiMajorAxis, double flattening);
    void    SetEllipsoid(int ID);
    double  GetEllipsoidDistance(double lat1, double lon1, double lat2, double lon2);
    double  GetVincentyDistance(double lat1, double lon1, double lat2, double lon2);
    double  GetGreatCircleDistance(double lat1, double lon1, double lat2, double lon2);
    double  GetApproximateDistance(double lat1, double lon1, double lat2, double lon2);
    double  GetDistanceAlongMeridian(double lat1, double lat2);
    double  GetDistanceAlongParallel(double lat, double lon1, double lon2);
};
```

Even though users can use all public methods, users can obtain ellipsoid geodesics by using the SetEllipsoid() and GetEllipsoidDistance() methods. The former is used for setting up ellipsoid parameters, and the latter for

calculating a geodesic. SetEllipsoid() should be used before using GetEllipsoidDistance(). This class supports 55 predefined ellipsoids. In SetEllipsoid(), the ID and parameters are as follows in Table 1 (Hooijberg, 1997). If users want to use their own ellipsoid, they can define an ellipsoid using either SetEllipsoid\_ab() or SetEllipsoid\_af() method instead of using the SetEllipsoid() method.

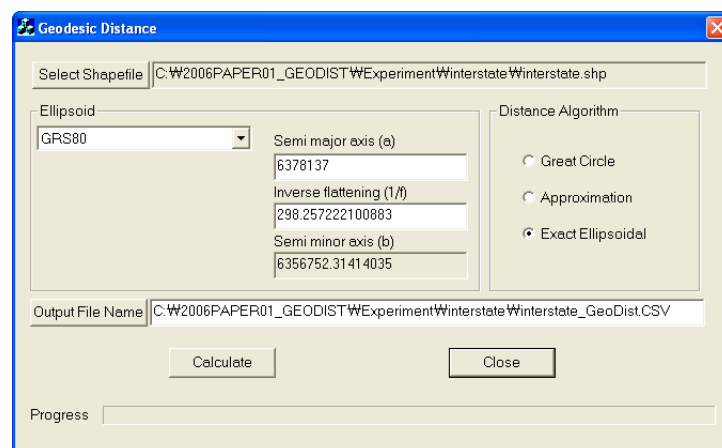
**Table 1. Ellipsoid ID and parameters that are used in the SetEllipsoid() method.**

ID	Name	InverseFlattening	SemiMajorAxis
0	Airy 1830	299.3249646	6377563.396
1	Airy modified 1965	299.3249646	6377340.189
2	Andrae 1876	300	6377104.43
3	APL 4.5 1965	298.25	6378137
4	Australian National (IAU 65)	298.25	6378160
5	Bessel 1841	299.1528129	6377397.155
6	Bessel (Namibia)	299.1528129	6377483.865
7	Bessel NGO 1948	299.1528	6377492.018
8	Clarke 1858	294.26	6378360.706
9	Clarke 1866	294.9786982	6378206.4
10	Clarke 1880 DoD	293.465	6378249.145
11	Clarke 1880 IGN	293.4660213	6378249.2
12	Clarke 1880 G	293.465	6378249.145
13	Clarke 1880 Arc Modified	293.4663076	6378249.145
14	Clarke 1880 Palestine	293.4663077	6378300.79
15	Danish National Grid	300	6377103.965
16	Delambre 1810	311.5	6378428
17	Engelis 1985	298.2566	6378136.05
18	Everest 1830 (India)	300.8017	6377276.346
19	Everest 1956 (India)	300.8017	6377301.243
20	Everest 1948 (W. Malaysia, Singapore)	300.8017	6377304.063
21	Everest 1969 (West Malaysia)	300.8017	6377295.664
22	Everest (Pakistan)	300.8017	6377309.613
23	Everest (Brunei, East Malaysia)	300.8017	6377298.556
24	Fisher 1960 / Mercury	298.3	6378166
25	Fisher 1960 Modified	298.3	6378155
26	Fisher 1968	298.3	6378150
27	Fisher 1968 (MMD)	298.2471674	6378160
28	GEM 10 1979	298.257	6378139
29	GRS67	298.2471675	6378160
30	GRS80 (or "New International")	298.2572221	6378137
31	Hayford 1909	297	6378388
32	Helmert 1906	298.3	6378200
33	Hough 1960	297	6378270
34	IAG 1975	298.257222	6378140
35	Indonesian 1974 (GRS 67)	298.2471675	6378160
36	International 1924	297	6378388
37	Kaula 1961	298.24	6378163
38	Krassovsky 1940	298.3	6378245
39	Maupertuis 1738	191	6397300

40	NWL-9D 1965	298.25	6378145
41	NWL-10D	298.26	6378135
42	Plessis 1810	308.64	6376523
43	Shdanov 1893	299.7	6377717
44	SGS85	298.257	6378136
45	SGS90	298.257	6378136
46	South American 1969 (GRS67)	298.25	6378160
47	Struve	294.73	6378297
48	Walbeck 1817	302.78	6376896
49	War Office (McCaw) 1924	296	6378300.58
50	WGS60	298.3	6378165
51	WGS66	298.25	6378145
52	WGS72	298.26	6378135
53	WGS84	298.2572236	6378137
54	WGS84 (G730)	298.2572012	6378137

As a largely used file format in GIS community, a shapefile stores the geometry for a feature as a shape comprising a set of vector coordinates. A shapefile format supports various vector geometries such as Point, PolyLine, Polygon, MultiPoint, PointZ, PolyLineZ, PolygonZ, MultiPointZ, PointM, PolyLineM, PolygonM, MultiPointM, and MultiPatch (ESRI, 1998). GEODIST application calculates geodesic distances for PolyLine, PolyLineZ, and PolyLineM features. A PolyLine is an ordered set of vertices that consists of one or more parts. A part is a connected sequence of two or more points. Parts may or may not be connected to one another. Parts may or may not intersect one another. PolylineZ and PolylineM features contain elevation and measurement in each point of a line, respectively. In order to read line features in the shapefile format, Warmerdam's Shapefile C Library (Warmerdam, 1999) was used in GEODIST.

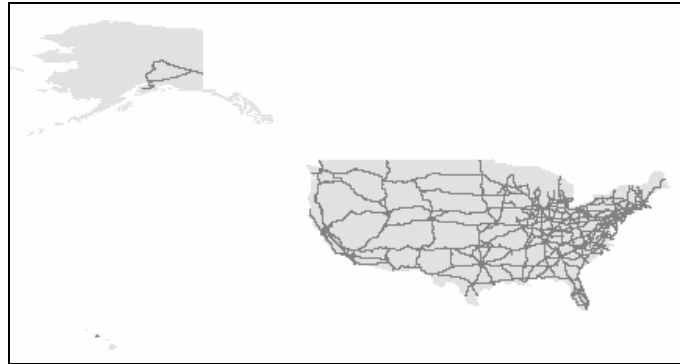
Figure 2 shows the GEODIST's user interface. Users can select a shapefile using the Select Shapefile button. After defining an ellipsoid and a distance algorithm, a user can obtain geodesic distance by clicking the Calculate button. Output is saved on a comma separated file which has the CSV file extension, which contains two columns – "ID" and "GeoDist." Geodesic distances are written in the GeoDist field. The ID column is sequentially numbered using the FID column in the original shapefile for later join to the original shapefile.



**Figure 2.** User Interface of GEODIST.

## DISTANT MEASUREMENT

GEODIST application was tested with a transportation dataset and National Hydrographic Dataset (NHD). The transportation dataset contains the Interstate highways (Figure 3) in the shapefile format (ESRI, 2005). The ESRI dataset contains distance information in the DIST\_KM field. Using GEODIST, the geodesic distances of 689 road features were calculated. Then, the difference between the DIST\_KM field and GeoDist field was calculated and analyzed. The root mean square error of the difference was 12.495km. The maximum difference was 158.377 km in I-80 followed by 122.672km in I-40 (Table 2). In average, 0.96% of difference was found. As a whole, even-numbered east-and-west highways showed larger differences than odd-numbered north-and-south highways.

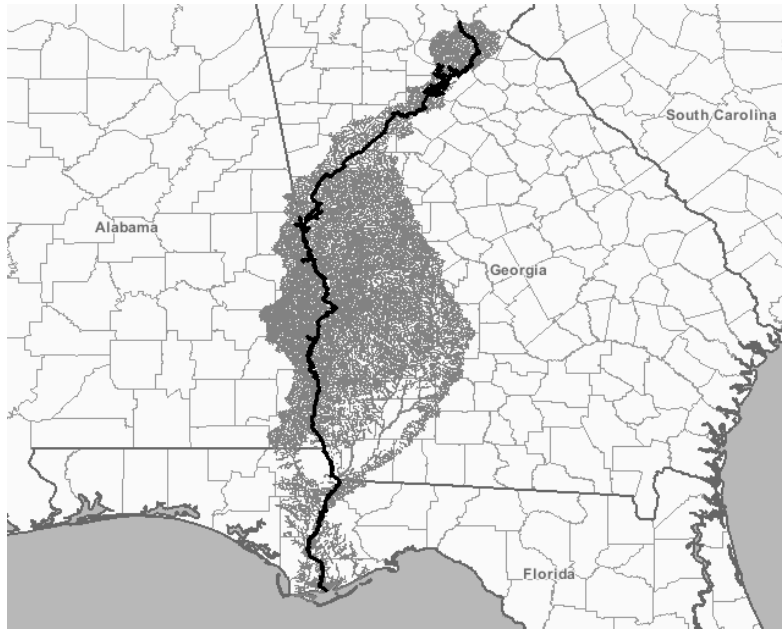


**Figure 3.** Interstate highways in geographic coordinate system

**Table 2.** Top ten longest Interstate highways and distance differences.

Interstate	Geodesic Distance	DIST_KM	Difference	% Difference
15	2295.908	2270.041	25.867	1.1
94	2228.075	2281.036	52.961	2.3
20	2450.308	2372.369	77.939	3.3
75	2809.287	2800.253	9.034	0.3
95	2929.660	2892.339	37.321	1.3
70	3397.577	3288.815	108.762	3.3
10	3913.797	3809.595	104.201	2.7
40	3967.388	3844.716	122.672	3.2
90	4421.431	4301.005	120.426	2.8
80	4579.689	4421.312	158.377	3.6

Another experiment was performed with the 1:100 000-scale National Hydrography Dataset (NHD). The NHD is a comprehensive set of digital spatial data that contains information about surface water features such as lakes, ponds, streams, rivers, springs and wells (USGS, 2005). Because of the large volume of the NHD datasets, a regional-scale river basin was selected and analyzed. Specifically, the Chattahoochee and Apalachicola river basin (i.e. NHDM0313 in NHD classification system) was selected in this experiment (Figure 4).



**Figure 4.** The Chattahoochee and Apalachicola river basin (NHDM0313)

The original hydrography data comes with the NHDFlowLine layer. The layer was dissolved with the GNIS\_Name field and the sum of the LengthKM field was calculated for each river. The sum field was compared with the calculation result from GEODIST. Results showed the average difference of 0.24% with the standard deviation of 0.43. In the Rockmore and Irvin creeks the differences were greatest with 4.62% and 4.72%, respectively.

## SUMMARY

In this study a GIS utility, GEODIST, was developed for the calculation of geodesic distances. GEODIST uses a shapefile format as input and a CSV file format as output for easy coupling with GIS packages. GEODIST is composed of three components – distance calculation (CGeodesic class), shapefile handling (shapelib.dll), and user interface (CGeoDist class). GEODIST calculates geodesic distances on the ellipsoidal surface using Vincenty's formulae. Experimental results on highways and rivers show that GEODIST is very effective in calculating accurate distances of GIS features in global and regional scales. The CGeodesic class in C++ also allows users to develop further applications by using two methods -- SetEllipsoid() and GetEllipsoidDistance(). The class also supports 55 predefined ellipsoids and additional user-defined ellipsoids. Results also showed that a higher accuracy can be expected by using a three dimensional geodesic calculation algorithm.

## REFERENCES

- Bomford, G., 1971. *Geodesy*, 3<sup>rd</sup> ed., Oxford University Press, London, U.K. 731pp.
- Defense Mapping School, 1973. *Basic Geodesy: Student Text*. Defense Mapping School, Fort Belvoir, Virginia. 40p.
- ESRI (Environmental Systems Research Institute), 1998. ESRI Shapefile Technical Description. URL: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. 28pp.
- ESRI (Environmental Systems Research Institute), 2005. *ESRI Data - ArcGIS Desktop 9.1*. Redland, California.
- Ewing, C. E. and Mitchell, M. M., 1970. *Introduction to Geodesy*. American Elsevier Publishing Company, Inc., New York. 304pp.
- Hooijberg, M., 1997, *Practical Geodesy Using Computers*. Springer, Berlin, 308pp.
- Intergraph, Inc., 2003. Common Applications of Geodesy and Map Projections, URL: <http://support.intergraph.com/TechNotes/mapproj.asp>.

Rainsford, H.F., 1955. Long Geodesics on the Ellipsoid. *Bulletin Geodesique*, 37, 12-22.  
USGS (U.S. Geological Survey), 2005. The National Hydrography Dataset. URL: <http://nhd.usgs.gov/index.html>.  
Vincenty, T., 1975. Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of nested Equations. *Survey Review XXII*, 176: 88-93.  
Warmerdam, F., 1999, Shapefile C Library V. 1.2. URL: <http://shapelib.maptools.org/>.