

Transformation of coordinates from local to global coordinate system and visualization of data by using Google Earth

Nenad Berić, Miro Govedarica and Đorđe Pržulj

University of Novi Sad
Faculty of Technical Sciences, Novi Sad
bericnenad@gmail.com

Abstract – Detail surveys measured on the local level use local coordinate systems. To make the data obtained from such surveys compatible with Google Earth, a geodetic transformation is required – from local Gauss-Krüger coordinate system to global WGS 84 coordinate system. Goal of this paper is to describe a computer program developed to generate KML document from surveying data, so that it can be visualized with Google Earth.

Keywords - Google Earth, KML, GIS, Gauss-Krüger, WGS 84

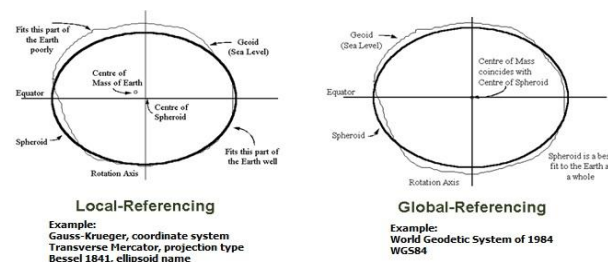
I. INTRODUCTION

Google Earth is a free computer program which uses a virtual globe to map the Earth using satellite and aerial imagery. Google Earth is a virtual globe that represents the whole planet and uses global GPS coordinate system WGS 84. On the local level all data obtained from land survey use local (e.g. Gauss-Krüger) coordinate system. It is essential to convert data from local to global coordinate system. Doing so will enable user to view their data in any application which is compatible with KML [1] format. This application will use Google Earth [2] for the visual representation of exported objects.

II. BACKGROUND

A. Coordinate Systems

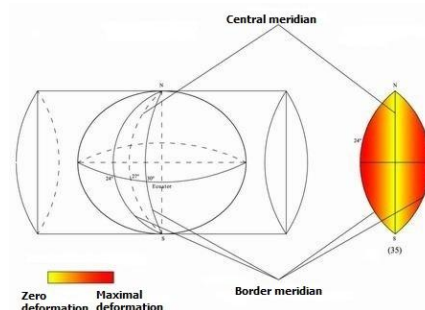
Data can be classified into two broad categories: local referencing data and global referencing data. A local referencing datum is a datum which best approximates the size and shape of a particular part of the Earth's sea-level surface. A global referencing datum best approximates the size and shape of the Earth as a whole. These data is not intended to be used as a good approximation of any particular part of the Earth. Rather, their application lies in mapping projects that are more global in nature (Picture 1).



Picture 1, difference between local and global referencing

In Serbia local coordinate system utilizes Gauss-Krüger projection. Serbia has direction north to south what makes it ideal for this type of projection (Picture 2).

Also known as Transverse Mercator, this projection is similar to the Mercator except that the cylinder is longitudinal along a meridian instead of the equator. The central meridian is placed on the region to be highlighted. This centering minimizes distortion of all properties in that region. This projection is best suited for land masses that stretch north-south. The Gauss-Krüger (GK) coordinate system is based on the Gauss-Krüger projection.



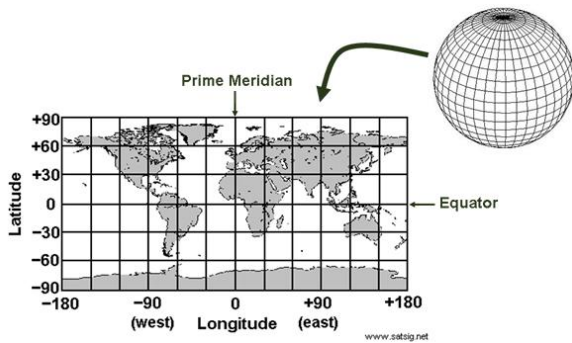
Picture 2, Gauss-Krüger projection

The country is divided in three zones each three degrees wide, roughly from 19 to 23 degrees longitude (Picture 3).



Picture 3, Gauss-Krüger projection for Serbia with zone division

The World Geodetic System datum is the only world referencing system in place today (Picture 4). WGS 84, the latest revision of World Geodetic System, is also the default standard datum for coordinates used in recreational and commercial GPS units. Google Earth uses GPS coordinate system WGS 84 for latitude and longitude and Earth Gravitational Model 1996 (EGM96) for height. WGS 84 is a coordinate system based on the GRS 80 (Geodetic Reference System 1980) ellipsoid, an ellipsoid which approximates the Earth as a whole in the most detail.



Picture 4, World Geodetic System layout

In order to transform coordinates from regional surveying points to WGS Helmert transformation can be used.

B. Transformation algorithm

The Helmert transformation (named after Friedrich Robert Helmert, 1843–1917, also called a seven-parameter transformation) is a transformation method within a three-dimensional space. It is frequently used in geodesy to produce distortion-free transformations from one datum to another using:

$$X_T = C + \mu RX$$

Where:

- X_T is the transformed vector
- X is the initial vector

The parameters are:

- C is translation vector, it contains the three translations along the coordinate axes
- μ is scale factor, which is dimensionless, and as it is usually expressed in ppm, it must be divided by 1,000,000
- R is rotation matrix. It consists of three axes (small rotations around the coordinate axes) r_x , r_y , r_z . The rotation matrix is an orthogonal matrix. The rotation is given in radians

The coordinates of a reference system B are derived from reference system A by the following formula:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^B = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} + (1 + s \times 10^{-6}) \cdot \begin{bmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^A \quad [3]$$

Or for each single parameter of the coordinate:

$$\begin{aligned} X_B &= c_x + (1 + s \times 10^{-6}) \cdot (X_A - r_z \cdot Y_A + r_y \cdot Z_A) \\ Y_B &= c_y + (1 + s \times 10^{-6}) \cdot (r_z \cdot X_A + Y_A - r_x \cdot Z_A) \\ Z_B &= c_z + (1 + s \times 10^{-6}) \cdot (-r_y \cdot X_A + r_x \cdot Y_A + Z_A) \end{aligned}$$

Projection parameters for Serbia		
Gauss-Krüger projection zone number 7		
Parameter	Symbol	Value
Semi major axis	a	6377397.155m
Inverse flattening	1/f	299.1528125448
X-axis rotation	R_x	0°00'04.910687''
Y-axis rotation	R_y	0°00'03.003308''
Z-axis rotation	R_z	-0°00'11.094034''
X-axis translation	Δx	-534.787m
Y-axis translation	Δy	-133.682m
Z-axis translation	Δz	-501.482m
Scale factor	R_m	-1.15673966 ppm

III. COMPUTER PROGRAMMING

The computer program was developed using Java, a modern object-oriented programming language. Java is portable, meaning programs developed with the Java

programming language can be run on any computer that supports the Java platform. Java is supported by all major PC operating systems, as well as many web browsers, mobile internet devices and mobile [4].

This program was developed using integrated programming environment Oracle JDeveloper 10.1. [5] This environment was chosen for development because it covers full development lifecycle: design, coding, debugging and optimization. For database manipulation programming environment of choice was Oracle SQL[6] Developer. All data that the program is using is stored in Oracle 9i database. Use of development tools and database management system from one software manufacturer ensured compatibility and fast and relatively trouble free development of application.

The computer program has two functions. The first is to generate polygon that represents single cadaster parcel. User chooses cadastral parcel from the list of all parcels for the given cadastral municipality. When the chosen parcel is marked user clicks the button “Geo-complex export” (Picture 5).

Parcel ID	Status	Area	Other
01040001	1	15	104
01040002	1	15	104
01040003	1	15	104
01040004	1	15	104
01040005	1	15	104
01040006	1	15	104
01040007	1	15	104
01040008	1	15	104
01040009	1	15	104
01040010	1	15	104
01040011	1	15	104
01040012	1	15	104
01040013	1	15	104
01040014	1	15	104
01040015	1	15	104
01040016	1	15	104
01040017	1	15	104
01040018	1	15	104
01040019	1	15	104
01040020	1	15	104
01040021	1	15	104
01040022	1	15	104
01040023	1	15	104
01040024	1	15	104
01040025	1	15	104
01040026	1	15	104
01040027	1	15	104
01040028	1	15	104
01040029	1	15	104
01040030	1	15	104

Picture 5, list of geodetic parcel's for one municipali

Once the action is invoked, the program generates KML file and opens it with Google Earth (Picture 6).



Picture 6, graphic representation of exported geodetic parcel in Google Earth

The second function is to generate polygon that represents cadastral municipality. This action is invoked by clicking on the “Cadastral municipality export” button, when desired cadastral municipality is selected from the list (Picture 7).

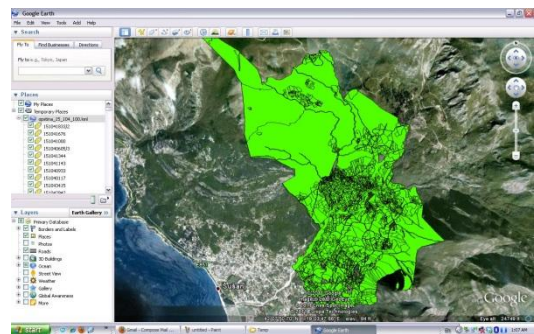
Municipality	Area	Other
0104	104	104
0105	105	105
0106	106	106
0107	107	107
0108	108	108
0109	109	109
0110	110	110
0111	111	111
0112	112	112
0113	113	113
0114	114	114
0115	115	115
0116	116	116
0117	117	117
0118	118	118
0119	119	119
0120	120	120
0121	121	121
0122	122	122
0123	123	123
0124	124	124
0125	125	125
0126	126	126
0127	127	127
0128	128	128
0129	129	129
0130	130	130

Picture 7, list of cadastral municipalities

Every cadastral municipality is made of a number of parcels that often can number several thousands. This can result in extremely big number of coordinate transformations. Implementing progress bar is absolutely necessary, so user can be informed about process progress.

The result of these actions is a number of single polygons that together represents the whole municipality (Picture 8).

In the tree structure on the Google Earth user interface, user can choose any parcel or simply can click on the graphical representation in the main window of Google Earth users interface.



Picture 8, export of all geodetic parcels for chosen municipality and graphic representation in Google Earth

The result of both actions is creation of a file in the file system of the user's computer.

Both functions use the input data to create a KML file, which includes the new WGS 84 (transformed) coordinates and formatting details. Google Earth uses the KML files to superimpose the data on top of its aerial imagery.

A. Java classes of interest used in the application

This class is responsible for the actual coordinate system transformation. Input parameters are coordinates in Gauss-Krüger system. After transformation output parameters are coordinates in WGS 84 system.

Once transformed these values are written in as XML tags using DocumentBuilderFactory class. This class defines a factory API that enables applications to obtain a parser that produces DOM object trees from XML documents. [6]

The `COORD` class encapsulates all methods related to Earth coordinate transformations. It can transform Earth coordinates between a huge number of Earth data, and it can transform from/to Gauss-Krüger coordinates. [7]

`SwingWorker` is an abstract class to perform lengthy GUI-interacting tasks in a dedicated thread. [8]

B. Future enhancements

There are two areas on which current application can be improved. The first is the interaction with user, the second is data representation.

User interface which would enable configuration of colors for cadastral objects that are generated and displayed could be implemented.

Second, if suitable data can be obtained, KML multi-geometry polygons can be generated. These polygons could represent not only the outline of cadastral parcel; they could represent built objects on it and other major characteristics.

Additional information can be displayed using KML balloon in Google Earth; for example, the size of parcel, name of the owner and other information of interest.

IV. CONCLUSION

WGS is a globally accepted coordinate system readily available to wide range of users. On the other hand, Gauss-Krüger coordinate system is used mostly by small number of expert users for specific purposes by specific software. Therefore this software bridges the gap between these two groups.

KML format, which this software uses for generation documents, is used by a number of different GIS software

solutions so it can be easily used for data exchange between them.

This program is not intended for precision use. It enables users to quickly see the cadastral objects and relations of them to other objects as well as relation to the terrain in Google Earth or other compatible KML software.

ACKNOWLEDGEMENTS

I would like to thank to my colleagues: Mirko Raca, Ivan Lukić and Luka Ranisavljević for their help with programming in Java language. Also special thanks go to Đorđe Pržulj for his guidance and patience while working on this project.

REFERENCE

- [1] <http://earth.google.com/support/bin/static.py?page=guide.cs&guide=22373&topic=23747&answer=148118> [accessed 09.09.2011]
- [2] <http://code.google.com/apis/earth/documentation/> [accessed 09.09.2011]
- [3] <http://www.linz.govt.nz/geodetic/conversion-coordinates/geodetic-datum-conversion/datum-transformation-equations/index.aspx> [accessed 09.09.2011]
- [4] Flanagan, D., 2005. Java in a Nutshell. 5th ed., Sebastopol: O'Reilly.
- [5] <http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html> [accessed 09.09.2011]
<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html> [accessed 09.09.2011]
- [6] <http://download.oracle.com/javase/1.4.2/docs/api/javax/xml/parsers/DocumentBuilderFactory.html> [accessed 09.09.2011]
- [7] <http://javagps.sourceforge.net/javadoc/org/iiu/gps/COORD.html#convertToLatLng%28double,%20double,%20int%29> [accessed 09.09.2011]
- [8] <http://download.oracle.com/javase/6/docs/api/javax/swing/SwingWorker.html> [accessed 09.09.2011]