ALGORITHM AND APPLICATION DEVELOPMENT FOR PRECISE AND ACCURATE TRANSFORMATION OF LIDAR POINT CLOUDS INTO NATIONAL COORDINATE SYSTEMS OF ROMANIA USING OFFICIAL EQUATIONS AND QUASIGEOID MODEL

D. Ilie [1, 2, 3] *, O. L. Balotă [2, 3, 4], D. Iordan [2, 3, 4], P. S. Nicoară [2]

[1] Technical University of Civil Engineering of Bucharest, Doctoral School, 020396 Bucharest, Romania – daniel.ilie@phd.utcb.ro
[2] Prosig Expert SRL, 022131 Bucharest, Romania – (daniel.ilie, octavian.balota, daniela.iordan)@prosig.ro
[3] Romanian Society of Photogrammetry and Remote Sensing, 020396 Bucharest, Romania
[4] University of Agronomic Sciences and Veterinary Medicine of Bucharest, 011464 Bucharest, Romania – octavian_ttt@yahoo.com, iordandaniela5@gmail.com

Commission IV, WG IV/7

KEY WORDS: Coordinate transformations, Quasigeoid, Batch Processing, LIDAR, LAS, Stereographic 1970, Black Sea 1975.

ABSTRACT:

The LiDAR point clouds are usually processed in Universal Transvers Mercator projection. The transformation to a national coordinate system is frequently made with low accuracy with the help of the generic transformation implemented in the actual softwares. An accurate and precise transformation for LiDAR files in the national coordinate systems of Romania (planimetric system Stereographic 1970 and altimetric system Black Sea 1975) is not yet available. The National Center for Cartography (NCC) from Romania developed a software for precise transformation, but it works only for certain patterns of the text files and for a maximum number of points of about 1 million. Because of this, the use of point clouds in precision work was not possible, using only extracts of low-density grid points in text format. In this research we develop an algorithm which use the precise transformation of NCC to realise an accurate transformation of the LiDAR point clouds in the national coordinate systems. The algorithm is then implemented in an innovative software to transform the LiDAR *LAS files, using the common version 1.2. The software is a batch processing application, which it can process big LiDAR data without blocking. Moreover, the application is capable to apply with accuracy and precision the last published national quasigeoid to the LiDAR point files. In the end, the obtained LiDAR point cloud are more suitable to be used in any domain, because of the accurate and precise transformation in the Romanian coordinate systems.

1. INTRODUCTION

1.1 National context

The situation of Romanian highway infrastructure is probably one of the worst in the European Union. With a length of 942 km and another 382 km under construction, Romania still not have a complete highway to cross the country from West to East or from North to South (Development team of 130km.Ro, 2022). The railway network is more developed in terms of length but is very old and unmodernized. Another major transport route is the waterway of Danube, part of the Rhine-Main-Danube corridor (the Pan-European Corridor VII connecting two European ports: Rotterdam and Constanta). Even if one third (1075 km) of this major fluvial route is in Romania, the transportation through it is not so easy to achieve. The Romanian sector needs to be regularized, dredged, and digitalised in order to be a safety and a competitive transport route.

This is the context in which the LiDAR technology gets its real importance. The modernization projects or the design ones for transport infrastructures need a topographic support as detailed as possible, but also precise and accurate. The LiDAR technology provides the most comprehensive solution for a detailed topographic support. As concerning the inner precision but also the inner accuracy of the LiDAR point clouds, these are very suitable for the transport infrastructures projects.

1.2 Technical situation

First we will remind the difference between precision and accuracy, from the inner or the external perspective of view. The inner precision is defined as the deviation between different measurements of a single point (the degree of reproducibility). The inner accuracy is the deviation from measurements between every two points to the true value of the distance between these points (true relative position).

The external precision is given by the georeferenced measurements of a single point (in other words the inner precision combined with the errors of the method of georeferencing). The difference between absolute position and the real position of a dataset is defined as the external accuracy. In this article we will address only the issue of external data precision and external data accuracy, after the transformation in the national coordinate systems.

The transformation of the LiDAR points in different national coordinate systems can lead to a loose of the final (external) precision and accuracy. This is also the case of Romania where an accurate and precise transformation for LiDAR files in the national coordinate systems is not yet available.

Beginning with 2009 Romania adopted the European Terrestrial Reference System 1989 (ETRS89) but only for realisation of the geodetic national network and Pan-European projects. The

* Corresponding author

ETRS89 is defined as being identical with the International Terrestrial Reference System (ITRS) at epoch 1989.0 (National agency for cadastre and land registration, 2009). The ETRS89 is composed of the geodetic datum ETRS89, based on the GRS80 ellipsoid, with ellipsoidal geodetic coordinate system. Based on these considerations the system was utilised to develop the main national network of permanent GNSS stations, named ROMPOS. This GNSS network is now determined in the datum of the European Terrestrial Reference Frame (ETRF) 2000, at the epoch of 2000.0.

The official planimetric coordinate system used in Romania is Stereographic 1970. This is a system with an oblique azimuthal projection that preserves the angles and deforms the distances and the areas. The associated geodetic datum is Krasovski 1942 which is based on the ellipsoid Krasovski 1940. The vast majority of the actual geodetic works are made using GNSS technology with the help of the national GNSS network ROMPOS (Development team of rompos, 2022). The differences between the two ellipsoids lead to a loss of precision and accuracy when transforming to the national planimetric coordinate system. The implementation of a national projection based on the GRS80 ellipsoid is still at the beginning (Rus, 2021). To make things even more complicated, the official vertical coordinate system is the Black Sea 1975 (Edition 1990), a system with normal heights. The reference of the vertical system is defined by the surface of the quasigeoid (Vanicek et al., 2012) based on the sea mean level determined by the tide gauge instrument installed in the city of Constanța (The national center for cartography, 2021a). The geoid EGG97 was the base model for the national quasigeoid (Dragomir and Sorta, 2012). The quasigeoid was first modelled on a few gravimetric points and more levelling points. Even if it was improved over the years with more gravimetric and level determinations, the quasigeoid model needs improvements for a quality transformation (Tiberiu et al., 2013).

1.3 Deficiencies of existing applications

All the geodetic, photogrammetric or LiDAR measurements are georeferenced with the help of the GNSS network, ROMPOS. This requires the necessity of a precise and accurate transformation from ETRS89 to the national coordinate systems (Stereographic 1970, Black Sea 1975 ed. 1990). For points transformation, The National Center for Cartography developed an application called TransDatRo v4.06, but this is working just for text files.

There are also some global software solutions but the transformations in the Romanian national coordinate systems are inaccurate and sometimes with a loose of precision. Moreover, the national quasigeoid model is not implemented and sometimes is difficult to use it or even impossible.

1.4 The necessity of a LAS transforming application to the national coordinate systems

All the mentioned aspects lead to the need of an application that can transform precise and accurate the LiDAR file to Romanian the national coordinate systems. Considering the big size of the LiDAR data point clouds is required a development of an application that can manage an entire project. The application should use the official transformation parameters and equations. As a major consequence, this application will encourage the use of LiDAR techniques in the large infrastructure projects.

2. OFFICIAL TRANSFORMATION PARAMETERS

2.1 Romanian coordinating authority

Over the years, plenty of Romanian researchers wrote about the equation of transformations between ETRS89 and national systems (Avramiuc et al., 2009). The National Center for Cartography (NCC), which is under the authority of the National Agency for Cadastre and Land Registration (NACLR), is the coordinating authority for cartographic transformation. The NCC put together all the research and geodetic measurements and develop the digital transformation parameters. As a result, it was developed the TransDatRo application (which has reached the version 4.06). Another derived application is ShapeTransDatRo version 2.00, which is working just for shapefiles (The national center for cartography, 2021b).

The NCC has the responsibility to refine the transformation grids to improve the quality of transformations. To implement this requirement, the NCC organise national campaigns of gravimetric measurement, combined with levelling and GNSS measurements. After a campaign is finished, NNC update the distortion grid corrections for the Stereographic 1970 transformation and the grid for quasigeoid undulation.

2.2 Official TransDatRo Application

TransDatRo, the official application for transformation in the Romanian coordinate systems, are written in Java programming language. Because of the different datums, there are no invariant mathematical conversion. This implies that a transformation must be made to obtain Stereographic 1970 coordinates from ETRS89. The official TransDatRo application realise a transformation in phases, which is refined on a distortion grid correction. The distortion grid corrections for the planimetric projection Stereographic 1970 are stored in a file named ETRS89_KRASOVSKI42_2D.GRT. The transformation from ETRS89 (GRS80 ellipsoid) to the Stereographic projection 1970 is made in three steps (Development team of transdatro, 2021a):

- The conversion from ETRS89 (GRS80 ellipsoid) to the oblique stereographic projection on GRS80:

$$N_G = F_N + 2 \cdot R \cdot k_0[\sin\chi\cos\chi_0 - \cos\chi\sin\chi_0\cos(\Lambda - \Lambda_0)]/\beta \quad (1)$$
$$E_G = F_E + 2 \cdot R \cdot k_0 \cdot \cos\chi\sin(\Lambda - \Lambda_0)/\beta \quad (2)$$

where F_N and F_E is False North and False East (500000m)
R is the radius of the mean Gaussian sphere
k_0 is the known scale coefficient (0.99975)
χ, Λ are the conformal latitude and longitude of a point with has the geodetic coordinates φ, λ
χ_0, Λ_0 are the conformal latitude and longitude of the pole of projection ($\chi_0 = 46°, \Lambda_0 = 25°$)
β is the length of the meridian arc measured on the ellipsoid between the parallel of latitude 46 and the parallel of latitude φ of the considered point.

- The conversion from oblique stereographic projection (on GRS80 ellipsoid) to approximative rectangular coordinates in Stereographic 1970 using a 4 parameter Helmert transformation:

$$N' = Y_0 + N_G \cdot m \cdot \sin(R_z) + Y \cdot m \cdot \cos(R_z) \quad (3)$$
$$E' = X_0 + E_G \cdot m \cdot \cos(R_z) - Y \cdot m \cdot \sin(R_z) \quad (4)$$

where E_G and N_G is East and North calculated on GRS80
X_0 and Y_0 is the translation on East, respectively on North
m is the scale coefficient

R_z is the rotation around the Z axis.

- Obtaining the final Stereographic 1970 coordinates by adjusting the coordinates from phase two with the correction interpolated on the distortion grid:

$$N_{ST70} = N' + p_y(E', N') \quad (5)$$
$$E_{ST70} = E' + p_x(E', N') \quad (6)$$

where p_y and p_y are the corrections on each axis, given by a function (depending by the coordinates calculated on phase two) with bicubic spline interpolation (which smooth the interpolations that is made).

The quasigeoid undulations are stored in a grid file named EGG97_QGRJ.GRT. This file is used to obtain normal heights in the Back Sea 1975 coordinate system using the following relation (Development team of transdatro, 2021a):

$$H_{MN75} = h_{el} - \zeta(\varphi, \lambda) \quad (7)$$

where h_{el} is the ellipsoidal height,
$\zeta(\varphi, \lambda)$ is the predicted quasigeoid anomaly, obtained by interpolation on the grid, with the known geodetic coordinates (φ, λ) of the given point.

The NCC does not provide an information about the precision of the planimetric transformation. As concerning the transformation into normal height there are three types of quality depending on the development of the quasigeoid: ±8.5 cm for Bucharest area, ±10-12 cm for 12 counties (where the quasigeoid was updated with more measurements), and ± 15-50 for the rest of the country, 29 counties (Development team of transdatro, 2021b).

2.3 Application limitations

The main purpose for TransDatRo was to transform points coordinates directly or from text files. The application was not optimised to work with large amount of data such are the LiDAR point clouds. A single point transformation requires a lot of calculation phases, and this is a reason why the application crash when the text file exceeds a number of about one million points. Other crashes were reported when it has reached the virtual memory limit because of the large number of the points in file. Another limitation of the TransDatRo application is the lack of UTM to ETRS89 transformation. For Romania, the UTM zones are 34 North and 35 North.

The modelling of the national quasigeoid is still developing and runs in stages, according with the campaigns of gravimetric determinations, precise levelling with GNSS measurements. Periodically the official grid that define the quasigeoid is updated in the official application (Spiroiu et al., 2017). All geodetic works should be made in the national coordinate systems using the latest published version of TransDatRo. Therefore, the source code of the application and the correction grids are freely provided by NCC for further implementation and development (The national center for cartography, 2020). The programming code of TransDatRo has rich the version 1.04 and usually is not changing.

The main goal of the actual research is to make an algorithm and then to implement it in an application for transforming the LAS files. The application should use the official parameters and equations (from (1) to (7)) for the ETRS89 to national systems transformation, so the results could be approved by NCC in the following works.

3. NEW APPLICATION ALGORITHM

3.1 Algorithm logical scheme

The LiDAR data are processed in a planimetric projection, usually in Universal Transverse Mercator (UTM). Because of using the ROMPOS network, the datum of UTM is also ETRS89, with GRS80 ellipsoid. The main reason of using this projection is because it represents a conversion from ETRS89, and thus error free (Ogp surveying and positioning committee's geodetic, 2021). Therefore, the algorithm should convert from the UTM 34N/35N to ETRS89 and then transform the coordinates to the Romanian national coordinate systems. Because of the limitation of the source code of TransDatRo the big files should be split in small files so the application does not crash or freeze. The logical scheme of the transforming algorithm is presented in **Figure 1**.

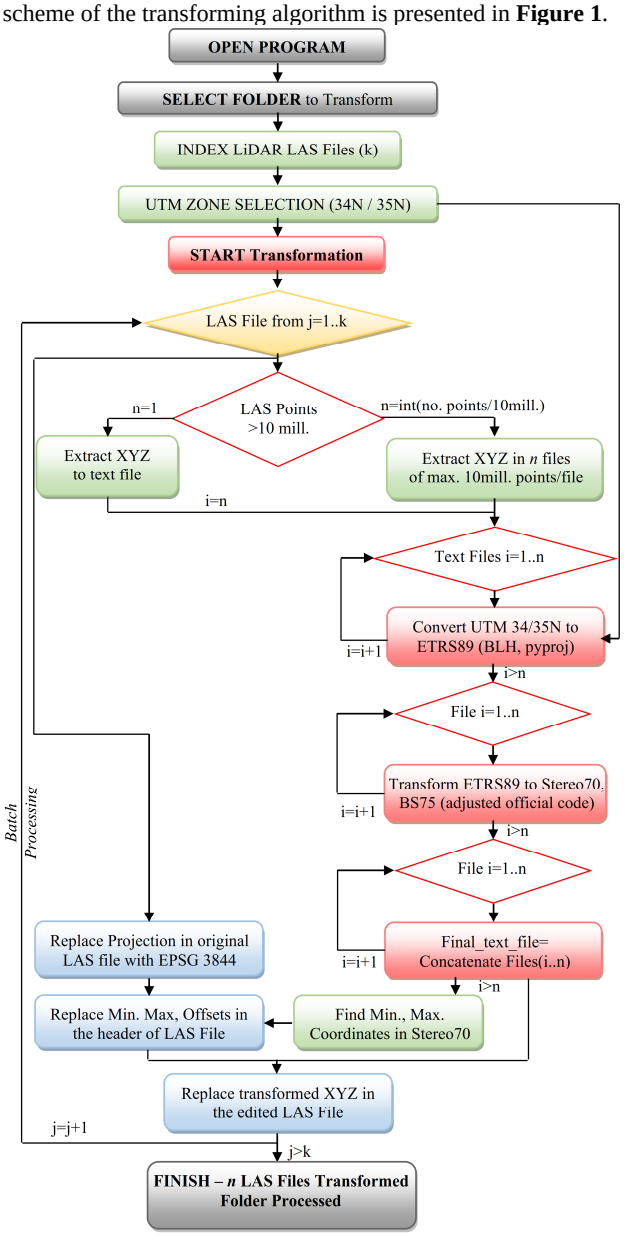

Figure 1. Logical scheme of the transforming algorithm

A LiDAR LAS file has a complex structure so it was developed an algorithm that does not write again the entire file. It was decided to replace the transformed coordinates in the initial file and the metadata that has to be changed. Therefore in the header of file the algorithm will replace the new projection (with the

EPSG[1] code for Stereographic 1970: 3844), offsets and the extents of the file (minimum and maximum). This approach will improve the processing time and will eliminate the possibility of the format inconsistency.

Because of the Java code from TransDatRo, we are forced to use text files. The limitation of the TransDatRo code needed to split the coordinates file so that no memory crash is encountered. The experimental tests show that the algorithm should split the big files at every 10 milions points. Each file will be converted, then transformed accordingly, and will be concatenated at the end. For smaller files this procedure will be omitted.

3.2 Batch processing and multiple sessions

The algorithm was constructed to process an entire folder for all the LAS files that are found there (**Figure 1**). Taking this into account, the algorithm was conceived to manage large amount of data without crashing or freezing. The algorithm will display messages after every important processing step, to ease the troubleshoot in case of errors. To improve the processing time, despite the limitations of the Java code (TransDatRo), the algorithm will run in separate temporary files so it can be runs in multiple working sessions. Even in this case, the main requirement remains unchanged: a smooth processing algorithm without crashing or freezing.

4. IMPLEMENTATION OF THE ALGORITHM

4.1 LAS file format version 1.2

To implement the algorithm described above, it was important to understand the format of a LAS file, which is elaborated by the American Society for Photogrammetry & Remote Sensing (ASPRS). There are four versions of LAS files, each of it with different point formats supported. For the application it was chosen the most used LAS format, version 1.2, and the point format number 3. This format include the most important point information: coordinates, intensity, return number, number of returns, scan direction flag, edge of flightline, classification, scan angle rank, point source ID, colours codes for each RGB band (Asprs board, 2009).

4.2 Used libraries

For programming it was used Python and Java languages. The main algorithm was written just in Python. For the implementation of the algorithm described above, the most important libraries that was used in Python script was:

- Os – a library for miscellaneous operating system interfaces. Provide a portable command way to manipulate diverse operating system procedures.
- Sys – a library that provide access to a specific parameters and variables maintain by the interpreter (Python).
- Tkinter – this is one of the most known Graphical user interfaces (Python software foundation, 2022a). It is used in the written code to generate the graphic display as is illustrated in **Figure 6**.
- Pathlib – this module offers object-oriented filesystem paths for different operating systems (Python software foundation, 2022b)
- Laspy – is a package for reading and modifying LiDAR LAS files, but also for creating new ones (Laspy community, 2021). This is the main library of the application script.
- Pyproj – is used to convert the coordinates from UTM 34N/35N to ETRS89 (Pyproj community, 2021), using the EPSG codes to identify the geodetic formulas (Ogp surveying and positioning committee's geodetic, 2009).
- Numpy – is a library for scientific computation that provides multidimensional arrays such as matrices, with fast routines for diverse arrays operations, mathematical functions, statistical analysis and much more (Numpy community, 2022).
- Others: subprocess, glob, datetime, time, threading.

There are also other packages that are used in the Java code to integrate in the Python algorithm (e.g., text, time, util, io, org.w3c.dom.ls.LSOutput).

4.3 Python programming

The main programming consists in Python implementation of the presented algorithm. One of the important steps is the UTM to ETRS89 conversion. After the UTM zone is selected, the conversion is made using the *transformer* module from *pyproj* library. The conversion parameters and function are called with the help of the EPSG code. For UTM zone 34N, based on ETRS89 ellipsoid, the algorithm uses the 25834 EPSG code (Ogp surveying and positioning committee's geodetic, 2000a). For UTM zone 35N, the function uses the 25835 EPSG code (Ogp surveying and positioning committee's geodetic, 2000b). After the conversion, the coordinates will be temporary stored in text files for the next phase. These coordinates are written with six decimals to minimalize the influence of approximation. In **Figure 2** is presented a part of the script that is used to convert the UTM coordinates to ETRS89.

```
with open(txt_path, "w") as f:
    print("zone", zone)
    transformer = Transformer.from_crs(UTM_zone_35N, ETRS_89)
    if int(zone) == 34:
        #print("zone 34")
        try:
            transformer = Transformer.from_crs(UTM_zone_34N, ETRS_89)

        except Exception as e:
            print("Error transformer", e)

    for i in range(len(x_array)):
        # print(x_array[i], y_array[i])
        try:
            latlong = transformer.transform(x_array[i], y_array[i])
            # print(transformer.description, transformer.from_crs)
        except Exception as e:
            print("Error transform", e)
        # print(latlong)
        latD = int(latlong[0])
        latM = int((latlong[0] - latD) * 60)
        latS = (latlong[0] - latD - latM / 60) * 3600
        longD = int(latlong[1])
        longM = int((latlong[1] - longD) * 60)
        longS = (latlong[1] - longD - longM / 60) * 3600
        # print(latD, latM, latS, longD, longM, longS)
        str_val = "{} {} {} {} {} {} {}".format(latD, latM, '%.6f' % latS,
                                                longD, longM, '%.6f' % longS, '%.5f' % z_array[i]) + "\n"

        try:
            # pass
            f.write(str_val)
            # f.close()
            #print("Exported file to ", txt_path)
        except Exception as e:
            print("Error", e)
```

Figure 2. UTM to ETRS89 conversion using pyproj library

The procedure illustrated in **Figure 2** runs for all the text files in which were extracted the coordinates from the LAS file. The next phase is the transformation to the national coordinates systems. This will call the Java script from TransDatRo, to realise the transformation from the converted ETRS89 coordinates. The Python code for this transformation is presented in **Figure 3**.

[1] European Petroleum Survey Group (now Oil and Gas Producers Surveying and Positioning Committee's Geodetic Subcommittee) created and maintain the most complete geodetic database projections

```
if nr_temp_files > 1:
    print("More than one temporary file.")
    for i in range(nr_temp_files):
        python_temp = input_las_file.rsplit(".", 1)[0] + "_{}_input.txt".format(i + 1)
        time = datetime.now()
        print(time.strftime("%H:%M:%S - "), "fisier python temp  care se converteste",
              python_temp)
        try:
            print("trying subprocess java call")
            print(r'"{}\java" "{}\giveNewValues.java" 1 "{}" "{}"'.
                  format(local_java_path, working_dir, python_temp, working_dir))
            subprocess.call(
                r'"{}\java" "{}\giveNewValues.java" 1 "{}" "{}"'.
                    format(local_java_path, working_dir, python_temp, working_dir),
                shell=True)
            os.remove(python_temp)
        except Exception as e:
            print("Failed java", e)
            time = datetime.now()
            message = "Error converting ETRS89 file to Stereo70."
            threading.Thread(
                target=self.text_area.insert(constants.END,time.strftime("%H:%M:%S - ") + message + '\n')).start()
        java_temp = python_temp.rsplit("_", 1)[0] + "_converted.txt"
        output_java_temp_list.append(java_temp)
        print("java_temp", java_temp)
        if os.path.isfile(java_temp):
            time = datetime.now()
            message = "The file {} has been converted from ETRS89 to Stereo70.".format(
                python_temp)
            threading.Thread(
                target=self.text_area.insert(constants.END,time.strftime("%H:%M:%S - ") + message + '\n')).start()
```

Figure 3. Python code for the transformation from ETRS89 to Stereographic 1970, Back Sea 1975

After the transformation is made the script will change the header information of the original LAS file, according to the new coordinates systems. The scrip for project change is illustrated in **Figure 4**. The existing projection stored in *GeoKey 3072* will be replaced with the EPSG code 3844 for Stereographic 1970. The script loop over the geo key entries (stored in groups of 4 values) and replace every information referred to the old projection.

```
def change_proj(src_path):
    replace_keys = {
        3072: 3844,  # set ProjectedCSTypeGeoKey to NAD83 / UTM 12N
        3076: 9001,  # set ProjLinearUnitsGeoKey to meters
        4099: 9001  # set VerticalUnitsGeoKey to meters
    }
    def update_geo_keys(path, lookup):
        with laspy.file.File(path, mode='rw') as f:
            for i, vlr in enumerate(f.header.vlrs):
                if vlr.record_id != 34735:
                    # print("vlr_record-id", vlr.record_id)
                    continue
                modified_body = []
                j = 0
                while j < len(vlr.parsed_body):
                    key_id, tag_loc, count, value = vlr.parsed_body[j:j + 4]
                    try:
                        value = lookup[key_id]
                    except KeyError:
                        pass
                    modified_body.extend((key_id, tag_loc, count, value))
                    j += 4
                vlr.parsed_body = modified_body
                break
            else:
                raise ValueError('no GeoKeyDirectoryTag found in file')
            f.header.save_vlrs()
    update_geo_keys(src_path, replace_keys)
```

Figure 4. Python code for projection change

The information about the geographic extends of the new files will be update in the header of the LAS file. In the end, the main script concatenates all the text files coming from the LAS file. The transformed coordinates are then replaced in the original LAS files, with the projection updated. In final step the temporary text files are deleted and the virtual memory are freed. Thus, the resulted file will have the same points information as the original LiDAR file (intensity, return number, number of returns, classification, point source ID, colours, etc.)

4.4 Java integration

The open source code from TransDatRo (v1.04), which is written in Java language, was needed to be adapted for calling in the main script. This was edited to accept the converted text files that contain points coordinates in ETRS89. A part of the changed Java code for integration in the application is shown in **Figure 5**.

```
public static void main(String[] args) throws NullPointerException, IOException {
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
    LocalDateTime read_now = LocalDateTime.now();
    BufferedReader reader;
    ArrayList<String> outString = new ArrayList<>();
    int choice = Integer.parseInt(args[0]);
    String next_st = args[1]; //input file
    String running_folder = args[2]; // folder the script is running in
    File file = new File(next_st); //input file
    String outFile = file.getAbsolutePath();
    int temp_outFile_ID;
    temp_outFile_ID = file.getAbsolutePath().lastIndexOf("_") - 1;
    outFile = outFile.substring(0, temp_outFile_ID+1)+"_converted.txt";
    ArrayList<String> fileReadLine = new ArrayList<>();
    try{
        reader = new BufferedReader(new FileReader(file));
        String line = reader.readLine();
        while (line!=null){
            fileReadLine.add(line);
            line = reader.readLine();
        }
        reader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    String NameFilegrdz= running_folder+"\\EGG97_QGR.GRT";
    String NameFilegrd_R= running_folder+"\\ETRS89_KRASOVSCHI42_2D.GRT";
    List<Double> EGG97_QGR_param = new ArrayList<>();
    Map<Integer, Collection<Double>> EGG97_QGR_shift_values = new HashMap<>();
    List<Double> ETRS89_KRASOVSCHI42_2D_param = new ArrayList<>();
    Map<Integer, Collection<Double>> ETRS89_KRASOVSCHI42_2D_shift_values = new HashMap<>();
    File EGG97_QGR_Obj = new File(NameFilegrdz);
    File ETRS89_KRASOVSCHI42_2D_Obj = new File(NameFilegrd_R);
```

Figure 5. Part of the changed Java code, adapted for application

At every application run, the Java script was changed to read and rewrite the database with the grid file find on the folder. As a major fact, when a new grid will be released (for Stereographic adjustments or for quasigeoid undulation) the user will need just to replace the existing grid in the application folder.

4.5 RoTLAS APP v1.62

The name of the new application was set to RoTLAS APP and is derived from **APP**lication for **Ro**manian **T**ransformation of **LAS** files. The graphic interface of the application is illustrated in **Figure 6**. The version 1.62 represents the final stage of a stable application with reliable results verified on the entire Romanian territory. The final application will run the transformation for all the LAS files identified in the selected folder. After the final tests were done, the scripts were packing as an executable application.

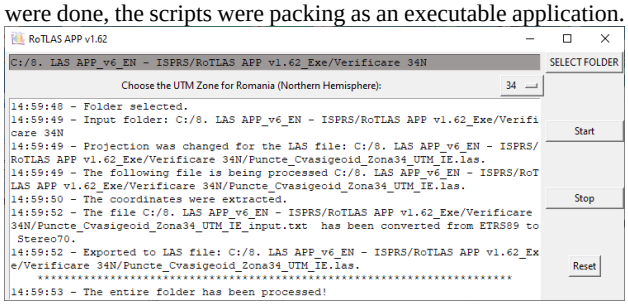

Figure 6. The graphic interface of RoTLAS APP v1.62

The copyrights of the RoTLAS APP are co-owned by the companies of the development team: Prosig Expert SRL and Terra Advanced Technologies SRL. The main scripts published here are free to use by other researchers. The license of the application is based on *Creative Commons* principle and it can be distributed as *Attribution* or as *Attribution-ShareAlike*.

5. RESULTS VALIDATION

5.1 Methods used

There are two directions for the application validation. The most important is the validation of the resulted transformation. The second validation is concerning about the analysis of the application functionality: batch processing, big data management, working time, multiple sessions.

The coordinates transformation was verified by comparison with other points transformed with TransDatRo v4.06. The transformation from UTM to ETRS89 was made using the converter of Inertial Explorer v8.80 from Novatel. For verifying the transformation from UTM 34N was used 96 points, and for

UTM 35N was used another 210 points. The points were uniformly distributed across the whole country as it can be seen in **Figure 7** (known as National Geodesic Network, class B).

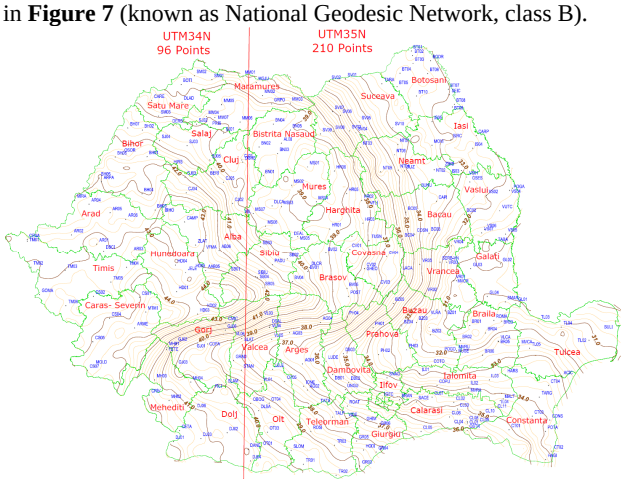

Figure 7. Points distribution used to validate the final results and the Romanian quasigeoid undulation

The actual quasigeoid undulation was extracted from the grid file in a text format, for next analysis. The variation of the quasigeoid undulation is expressed through contour lines in **Figure 7**.

5.2 Validation for UTM34N

Starting from the same coordinates in the UTM projection it was compared the results using the official transformation and the RoTLAS APP v1.62. In the **Table 1** are represented 10% of the points used for validation. However, the final statistic from **Table 1** are based on the whole data set results. The resulted differences are in fact residuals errors that are within 1mm.

INITIAL COORDINATES (Inertial Explorer v8.80, TransDatRo v4.06)				TRANSFORMED COORDINATES (RoTLAS APP v1.62)			DIFFERENCES (RoTLAS APP v1.62 - TransDatRo v4.06)				
NAME	North (m)	East (m)	Height (m)	North (m)	East (m)	Height (m)	dN (m)	dE (m)	dH (m)	2D error	3D error
ZLAT	514739.212	361840.757	429.665	514739.213	361840.757	429.665	0.001	0.000	0.000	0.001	0.001
STAN	364277.035	416512.987	382.556	364277.035	416512.987	382.556	0.000	0.000	0.000	0.000	0.000
SI01	648633.423	395277.618	208.933	648633.424	395277.618	208.933	0.001	0.000	0.000	0.001	0.001
MM01	716493.002	418443.273	274.879	716493.003	418443.273	274.879	0.001	0.000	0.000	0.001	0.001
HD03	431194.767	364340.584	624.374	431194.768	364340.584	624.374	0.001	0.000	0.000	0.001	0.001
FETE	388720.702	327210.644	597.242	388720.702	327210.644	597.242	0.000	0.000	0.000	0.000	0.000
CS05	398235.851	239064.579	211.934	398235.852	239064.579	211.934	0.001	0.000	0.000	0.001	0.001
CJ02	564388.470	405984.036	374.380	564388.470	405984.036	374.380	0.000	0.000	0.000	0.000	0.000
BH05	594833.851	246239.794	94.306	594833.852	246239.794	94.306	0.001	0.000	0.000	0.001	0.001
AR04	563683.199	234224.261	94.686	563683.200	234224.261	94.686	0.001	0.000	0.000	0.001	0.001
						Max. diff.	0.0010	0.0010	0.0010	0.0014	0.0017
						Min. diff.	0.0000	-0.0010	-0.0010	0.0000	0.0000
						Average	0.0007	0.0000	0.0000	0.0007	0.0008
						Median	0.0010	0.0000	0.0000	0.0010	0.0010
						Standard Dev.	0.0005	0.0003	0.0003	0.0005	0.0005
						RMSE	0.0008	0.0003	0.0003	0.0009	0.0009

Table 1. Validation statistics for transformation from UTM 34N

5.3 Validation for UTM35N

The validation of the transformation from UTM 35N to the national coordinates systems are done by the results centralized in **Table 2**. Only 10% of the points used in validation are shown, but the final statistics concerns all the points that was used.

INITIAL COORDINATES (Inertial Explorer v8.80, TransDatRo v4.06)				TRANSFORMED COORDINATES (RoTLAS APP v1.62)			DIFFERENCES (RoTLAS APP v1.62 - TransDatRo v4.06)				
NAME	North (m)	East (m)	Height (m)	North (m)	East (m)	Height (m)	dN (m)	dE (m)	dH (m)	2D error	3D error
ZIMB	608595.412	660341.221	439.679	608595.413	660341.221	439.679	0.001	0.000	0.000	0.001	0.001
VR03	493005.025	662236.635	269.168	493005.025	662236.635	269.168	0.000	0.000	0.000	0.000	0.000
VFLI	560218.676	567426.320	1289.918	560218.676	567426.320	1289.918	0.000	0.000	0.000	0.000	0.000
TATA	323014.548	508336.339	153.336	323014.548	508336.339	153.336	0.000	0.000	0.000	0.000	0.000
SV05	661975.396	557776.109	589.331	661975.396	557776.109	589.331	0.000	0.000	0.000	0.000	0.000
SIBIU	476126.158	433742.579	444.986	476126.159	433742.579	444.986	0.001	0.000	0.000	0.001	0.001
ROMA	423698.811	718642.103	22.304	423698.812	718642.103	22.304	0.001	0.000	0.000	0.001	0.001
PADU	490092.881	453733.573	594.873	490092.881	453733.573	594.873	0.000	0.000	0.000	0.000	0.000
NT05	602194.865	583303.633	414.306	602194.865	583303.633	414.306	0.000	0.000	0.000	0.000	0.000
MS05	524636.985	484247.010	348.534	524636.986	484247.010	348.534	0.001	0.000	0.000	0.001	0.001
LUDE	374908.430	517895.928	373.341	374908.430	517895.928	373.341	0.000	0.000	0.000	0.000	0.000
IL01	360258.275	629942.392	58.615	360258.276	629942.392	58.615	0.001	0.000	0.000	0.001	0.001
GR06	296164.819	580306.930	79.916	296164.819	580306.930	79.916	0.000	0.000	0.000	0.000	0.000
DSAL	418049.181	450708.527	503.987	418049.181	450708.527	503.987	0.000	0.000	0.000	0.000	0.000
DANE	269822.899	424476.277	123.955	269822.899	424476.276	123.955	0.000	-0.001	0.000	0.001	0.001
COSN	527650.188	623877.652	783.408	527650.189	623877.652	783.408	0.001	0.000	0.000	0.001	0.001
CL04	299786.198	677157.282	14.144	299786.198	677157.282	14.144	0.000	0.000	0.000	0.000	0.000
BZ01	435125.094	660306.472	143.919	435125.095	660306.472	143.920	0.001	0.000	0.001	0.001	0.001
BT06	719597.777	639426.907	137.420	719597.778	639426.907	137.420	0.001	0.000	0.000	0.001	0.001
BR03	422870.468	727924.934	15.159	422870.468	727924.934	15.160	0.000	0.000	0.001	0.000	0.001
BC04	521889.336	609359.852	573.147	521889.337	609359.852	573.147	0.001	0.000	0.000	0.001	0.001
						Max. diff.	0.0010	0.0010	0.0010	0.0014	0.0014
						Min. diff.	0.0000	-0.0010	-0.0010	0.0000	0.0000
						Average	0.0006	-0.0001	0.0000	0.0007	0.0007
						Median	0.0010	0.0000	0.0000	0.0010	0.0010
						Standard Dev.	0.0005	0.0003	0.0002	0.0005	0.0005
						RMSE	0.0008	0.0003	0.0002	0.0008	0.0009

Table 2. Validation statistics for transformation from UTM 35N

As expected the resulted differences are within 1mm and are in facts residuals errors of the transformation. For both transformation (34N/35N), the maximum difference found on a coordinate was 1mm.

6. RESULTS ANALYSIS AND COMPARISON TO OTHER APPLICATION

In this chapter the results of the new application will be subject to a comparison to the most known transformation applications such as Global Mapper, ArcGIS PRO and TerraScan. None of these application does not have implemented the official transformation to Romanian coordinate systems. For comparison was used the data set from the UTM 35N validation (210 points).

6.1 Global Mapper

Global Mapper is one of the most used software for conversion or transformation of diverse files. The transformation from UTM to Stereographic 1970 are fast but not so precise. This is because the software does not have implemented the planimetric correction grid. Partial results and the whole statistic are presented in **Table 3**.

INITIAL COORDINATES (Inertial Explorer v8.80, TransDatRo v4.06)				TRANSFORMED COORDINATES (Global Mapper v23.0 64-bit)			DIFFERENCES (Global Mapper-TransDatRo)				
NAME	North (m)	East (m)	Height (m)	North (m)	East (m)	Height (m)	dN (m)	dE (m)	dH (m)	2D Error	3D Error
ZIMB	608595.412	660341.221	439.679	608595.529	660340.881	439.678	0.117	-0.340	-0.001	0.360	0.360
VR03	493005.025	662236.635	269.168	493004.648	662236.576	269.163	-0.377	-0.059	-0.005	0.382	0.382
VFLI	560218.676	567426.320	1289.918	560218.305	567426.383	1289.931	-0.371	0.063	0.013	0.376	0.377
TATA	323014.548	508336.339	153.336	323014.343	508336.118	153.336	-0.205	-0.221	0.000	0.301	0.301
SV05	661975.396	557776.109	589.331	661974.890	557775.918	589.323	-0.506	-0.191	-0.008	0.541	0.541
SIBIU	476126.158	433742.579	444.986	476125.846	433742.364	444.982	-0.312	-0.215	-0.004	0.379	0.379
ROMA	423698.811	718642.103	22.304	423698.621	718641.933	22.296	-0.190	-0.170	-0.008	0.255	0.255
PADU	490092.881	453733.573	594.873	490092.594	453733.334	594.868	-0.287	-0.239	-0.005	0.373	0.374
NT05	602194.865	583303.633	414.306	602194.554	583303.648	414.248	-0.311	0.015	-0.058	0.311	0.317
MS05	524636.985	484247.010	348.534	524636.784	484247.018	348.535	-0.201	0.008	0.001	0.201	0.201
LUDE	374908.430	517895.928	373.341	374908.370	517895.693	373.336	-0.060	-0.235	-0.005	0.243	0.243
IL01	360258.275	629942.392	58.615	360257.806	629942.322	58.594	-0.469	-0.070	-0.021	0.474	0.475
GR06	296164.819	580306.930	79.916	296164.451	580306.905	79.912	-0.368	-0.025	-0.004	0.369	0.369
DSAL	418049.181	450708.527	503.987	418049.313	450708.460	503.987	0.132	-0.067	0.000	0.148	0.148
DANE	269822.899	424476.277	123.955	269822.992	424475.718	123.969	0.093	-0.559	0.014	0.567	0.567
COSN	527650.188	623877.652	783.408	527649.874	623877.736	783.404	-0.314	0.084	-0.004	0.325	0.325
CL04	299786.198	677157.282	14.144	299785.884	677157.281	14.139	-0.314	-0.001	-0.005	0.314	0.314
BZ01	435125.094	660306.472	143.919	435124.750	660306.497	143.915	-0.344	0.025	-0.004	0.345	0.345
BT06	719597.777	639426.907	137.420	719597.897	639426.635	137.418	0.120	-0.272	-0.002	0.297	0.297
BR03	422870.468	727924.934	15.159	422870.340	727924.743	15.150	-0.128	-0.191	-0.009	0.230	0.230
BC04	521889.336	609359.852	573.147	521889.110	609360.043	573.150	-0.226	0.191	0.003	0.296	0.296
						Max. diff.	0.417	0.277	0.057	1.110	1.110
						Min. diff.	-1.098	-0.806	-0.058	0.042	0.042
						Average	-0.227	-0.118	0.002	0.342	0.342
						Median	-0.236	-0.089	0.000	0.315	0.315
						Standard Dev.	0.226	0.183	0.013	0.181	0.181
						RMSE	0.320	0.218	0.013	0.387	0.387

Table 3. Transformation results with Global Mapper v23.0

For a better understanding, **Figure 8** expresses 3D error size statistics for the entire National Geodesic Network, class B.

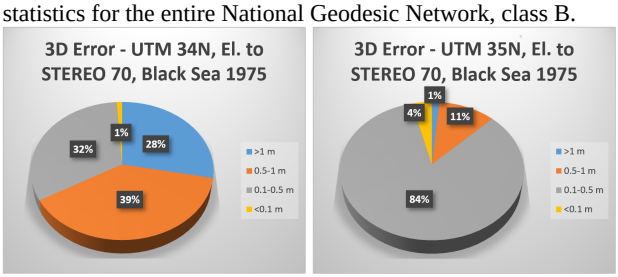

Figure 8. Transformation statistics for 3D errors

Despite the unsatisfactory results, Global Mapper offer the best transformation for the LAS files among the known softwares. However, the quasigeoid undulation are not easy to apply. There is no direct application of the grid file and the user should make a custom 3D model (**Figure 7**) and subtract it from the LAS ellipsoidal heights. Moreover, the application is freezing during the quasigeoid undulation application to large LiDAR files.

6.2 ArcGIS PRO and combined solutions

Another very well-known application for its transformation is ArcGIS PRO (v2.9). But for Romania, these transformations are useless. For planimetric transformation the RMSE is about 122 meters, far away for any accepted purpose. An explanation of those results could be a wrong definition of the Stereographic projection or the lack of the Helmert transformation (second step

of the transformation). Also, there is an absence of vertical projection transformation or quasigeoid undulation application. For this reason, ArcGIS PRO v2.9 was evaluated just for planimetric transformation. The results are presented in **Table 4**.

POINT NAME	TRANSFORMED COORDINATES (ArcGIS PRO v2.9)			DIFFERENCES (ArcGIS PRO v2.9 - TransDatRo v4.06)				TRANSFORMED COORDINATES (Global Mapper 2D+Terra Scan 1D)			DIFFERENCES (GM, TS - TransDatRo)			
	North (m)	East (m)	Height (m)	dN (m)	dE (m)	dH (m)	2D Error	North (m)	East (m)	Height (m)	dN (m)	dE (m)	dH (m)	3D Error
ZIMB	608566.546	660223.084	472.821	-28.866	-118.137	33.142	121.612	608595.529	660340.881	439.678	0.117	-0.340	-0.001	0.360
VR03	492973.645	662119.056	300.765	-31.380	-117.579	31.597	121.694	493004.648	662236.576	269.162	-0.377	-0.059	-0.006	0.382
VFLI	560188.278	567307.112	1328.047	-30.398	-119.208	38.129	123.023	560218.305	567426.384	1289.922	-0.371	0.064	0.004	0.377
TATA	322980.083	508216.321	190.969	-34.465	-120.018	37.633	124.869	323014.343	508336.118	153.338	-0.205	-0.221	0.002	0.301
SV05	661946.611	557656.260	624.806	-28.785	-119.849	35.475	123.257	661974.890	557775.918	589.330	-0.506	-0.191	-0.001	0.541
SIBIU	476094.085	433621.018	486.935	-32.073	-121.561	41.949	125.721	476125.846	433742.364	444.982	-0.312	-0.215	-0.004	0.379
ROMA	423666.513	718525.515	53.661	-32.298	-116.588	31.357	120.979	423698.621	718641.933	22.300	-0.190	-0.170	-0.004	0.255
PADU	490061.115	453612.294	635.621	-31.766	-121.279	40.748	125.370	490092.594	453733.334	594.869	-0.287	-0.239	-0.004	0.374
NT05	602165.292	583184.556	449.705	-29.573	-119.077	35.399	122.694	602194.554	583303.649	414.283	-0.311	0.016	-0.023	0.312
MS05	524605.965	484126.419	387.993	-31.020	-120.591	39.459	124.517	524636.784	484247.018	348.535	-0.201	0.008	0.001	0.201
LUDE	374875.027	517775.961	409.097	-33.403	-119.967	35.756	124.530	374908.370	517895.693	373.337	-0.060	-0.235	-0.004	0.243
IL01	360224.419	629824.513	91.579	-33.856	-117.879	32.964	122.645	360257.806	629942.322	58.606	-0.469	-0.070	-0.009	0.474
GR06	296129.855	580188.372	117.087	-34.964	-118.558	37.171	123.606	296164.451	580306.905	79.917	-0.368	-0.025	0.001	0.369
DSAL	418016.585	450587.515	542.980	-32.596	-121.012	38.993	125.325	418049.313	450708.460	503.985	0.132	-0.067	-0.002	0.148
DANE	269787.662	424354.612	164.988	-35.237	-121.665	41.033	126.665	269822.992	424475.719	123.961	0.093	-0.558	0.006	0.566
COSN	527619.399	623759.491	817.827	-30.789	-118.161	34.419	122.106	527649.874	623877.736	783.403	-0.314	0.084	-0.005	0.325
CL04	299751.524	677040.386	49.948	-34.674	-116.896	35.804	121.930	299785.884	677157.281	14.137	-0.314	-0.001	-0.007	0.314
BZ01	435092.730	660189.061	174.971	-32.364	-117.411	31.052	121.790	435124.750	660306.497	143.916	-0.344	0.025	-0.003	0.345
BT06	719570.808	639308.230	170.742	-26.969	-118.677	33.322	121.703	719597.897	639426.635	137.419	0.120	-0.272	-0.001	0.297
BR03	422838.235	727808.485	46.704	-32.233	-116.449	31.545	120.828	422870.340	727924.743	15.163	-0.128	-0.191	0.004	0.230
BC04	521858.504	609241.564	608.994	-30.832	-118.288	35.847	122.240	521889.110	609360.043	573.153	-0.226	0.191	0.006	0.296
	Max. diff.			-26.393	-114.036	42.268	126.926				Max. diff.			
	Min. diff.			-36.469	-121.842	30.611	118.447				Min. diff.			
	Average			-31.752	-118.666	35.564	122.863				Average			
	Median			-32.021	-118.780	35.156	122.645				Median			
	Standard Dev.			2.341	1.832	3.031	1.773				Standard Dev.			
	RMSE			31.838	118.680	35.693	122.876				RMSE			

				-1.098	-0.806	-0.033	0.042
				-0.227	-0.118	0.001	0.342
				-0.236	-0.089	0.001	0.315
				0.226	0.183	0.008	0.181
				0.320	0.217	0.008	0.387

Table 4. Transformation results with ArcGIS PRO v2.9 and Global Mapper v23.0 combined with TerraScan

Because of the Global Mapper complicated way of quasigeoid undulation application, was assessed a combined method. The Global Mapper planimetric transformation was used as input in TerraScan for the vertical transformation. It is not ascertained a substantial improvement (**Table 4**), except the processing time (**Table 5**). It must be mentioned that TerraScan does not have implemented a planimetric transformation for Romania.

6.3 Big Data Management

The second validation of RoTLAS APP v1.62 functionality was evaluated for a large dataset of files (12 files, centralizing 365 mill. points, with an amount of 10GB). With the batch processing option, RoTLAS APP v1.62 transformed this data set in one running session. Another test was done with five running sessions for the divided dataset (approx. 2GB each). RoTLAS APP v1.62 running in multiple working sessions, and the implicated resources are shown in **Figure 9**.

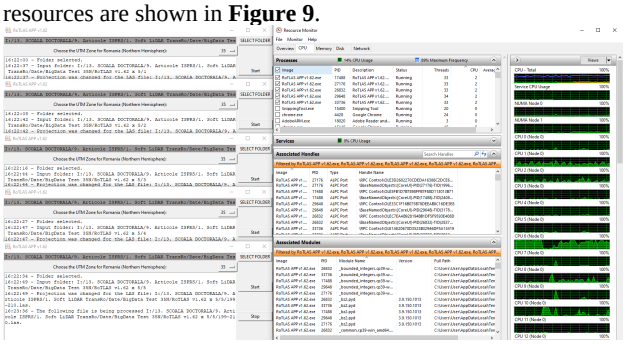

Figure 9. Running multiple sessions of RoTLAS APP v1.62 and the implicated resources

As it can be observed, RoTLAS APP v1.62 uses about 30 threads and two CPU (Central Processing Unit) when a session is running. The memory allocation is dynamic, taking into account other running processes. Thereby, the application made can run on any computer without freezing or crashing.

6.4 Processing speed

For faster data processing, the application RoTLAS APP v1.62 can run in multiple sessions as it can be seen in **Figure 10**. The example shown below represents the finish time of the big data processing task (10 GB). Another verification was about the format of the final exported LAS files. The test was done interchangeable with Global Mapper and TerraScan.

Figure 10. Running multiple sessions of RoTLAS APP v1.62 for the big data processing task (finish time)

Regarding the processing time, the best solution is the combined solution between Global Mapper and TerraScan. But RoTLAS APP v1.62 win at the category of the effective working time which is more important. The results concerning the processing time and the effective working time are centralised in **Table 5**.

Application	Processing Time (h:mm)	Effective Working Time (h:mm)	Observations
Global Mapper	1:06	0:39	Freeze and long procesing time for quasigeoid application. Imprecise planimetric transformation.
Global Mapper, Terra Scan	0:29	0:29	Improve the quasigeoid application (precision and working time). Imprecise planimetric transformation.
RoTLAS APP v1.62	3:20	0:01	Smallest effective working time. User can work at the same time on other projects.
RoTLAS APP v1.62 x 5 Sessions	0:44	0:05	Can run without freezing or crashing. User can work at the same time on other projects.
AVERAGE TIME	1:24	0:18	

Table 5. Results on the processing time of 10GB LAS files

7. CONCLUSIONS

The new developed application, RoTLAS APP v1.62, fills a gap in the Romanian geodetic transformations. RoTLAS APP v1.62 is the only one application for LiDAR LAS files which uses the current official equations and quasigeoid undulations. Moreover, this application can be easily updated just by replacing the corrections grid files, every time a new version is released.

Comparing with other transformation made with known softwares, this application improves the precision and the accuracy of the coordinate transformation. Considering as a reference the results obtained with the official TransDatRo, the best indicator for the precision of the transformations is the standard deviation. As it can be analysed in **Table 1** and **Table 2** the standard deviation for a 3D transformation is 0.5mm, much better than the Global Mapper transformation, with a standard deviation of 0.181m. As it was shown above, an imprecise transformation can affect the inner accuracy of the points. This is also the case of Global Mapper where an imprecise 2D transformation induce errors in the quasigeoid application, by applying imprecisely the quasigeoid grid. The maximum error is

not much (58mm), but for geodetic works every loose of precision is a problem. RoTLAS APP v1.62 resolve this issue by realising an accurate planimetric transformation. The best statistic parameter that proves this affirmation is RMSE, calculated against the values obtained with the official application for text files (predicted values). The statistical results (**Table 1** and **Table 2**) present a RMSE of 0.8mm for 2D transformation and a total RMSE of 0.9mm for 3D. Must be mentioned that resulted differences are in fact residuals errors that are within 1mm.

As regarding the big data management, the developed application was also evaluated on a project of about 100GB, and the results was as mentioned above: no freezing and no crashing. Because of the batch processing algorithm, but also of the multi session capability, RoTLAS APP v1.62 manage to process large amount of data in a competitive time. Moreover, the effective working time are the lowest, as it can be seen in **Table 5**. All those conclusions make RoTLAS APP v1.62 a unique application for LiDAR file transformation in the Romanian coordinate systems.

For future development and research purposes we intend to integrate the multi sessions capability in the core of the application. Furthermore, we aim to extend the capability of the application to transform LiDAR files from LAS version 1.3, LAS version 1.4 and from LAZ files. We are also searching for other researchers to help us integrate other transformation capabilities for different coordinate systems of other countries.

REFERENCES

ASPRS Board: LAS Specification Version 1.2, 2009.

Avramiuc, N., Dragomir, P. I., and Rus, T.: Algorithm for direct and inverse coordinate transformation between ETRS89 CRS and S-42 CRS, *RevCAD Journal of Geodesy and Cadastre*, 105-114, 2009.

Development Team of 130km.ro: List of Motorways in Romania: https://www.130km.ro/highways.html, last access: January 13, 2022.

Development Team of ROMPOS: User's Manual, 2022.

Development Team of TransDatRo: TransDatRO code source, 2021a.

Development Team of TransDatRo: User Manual, 2021b.

Dragomir, P.-I. and Sorta, V.: The situation in Eastern Europe about geoid/quasigeoid models determination, *RevCAD*, 13, 2012.

Laspy community: Laspy Documentation: https://pythonhosted.org/laspy/index.html, last access: January 16, 2022.

National Agency for Cadastre and Land Registration: Adoption in Romania of the European Terrestrial Reference System 1989, 2009.

NumPy community: NumPy User Guide, 2022.

OGP Surveying and Positioning Committee's Geodetic: ETRS89 / UTM zone 34N: https://epsg.io/25834, last access: January 15, 2022.

OGP Surveying and Positioning Committee's Geodetic: ETRS89 / UTM zone 35N: https://epsg.io/25835, last access: January 15, 2022.

OGP Surveying and Positioning Committee's Geodetic: EPSG Geodetic Parameter Relational Database-Dev. Op. Guide, 2009.

OGP Surveying and Positioning Committee's Geodetic: Coordinate Conversions and Transformations including Formulas, 2021.

Pyproj community: API Documentation Pyproj.Transformer: https://pyproj4.github.io/pyproj/stable/api/transformer.html, last access: January 15, 2022.

Python Software Foundation: Graphical User Interfaces with Tk: https://docs.python.org/3/library/tk.html, last access: January 16, 2022.

Python Software Foundation: Pathlib — Object-oriented filesystem paths: https://docs.python.org/3/library/pathlib.html#module-pathlib, last access: January 16, 2022.

Rus, T.: About the current state and problems of the reference and coordinate systems in Romania: https://www.revistaconstructiilor.eu/index.php/2021/07/01/ugr-despre-stadiul-actual-si-probleme-ale-sistemelor-de-referinta-si-coordonate-in-romania/, last access: January 13, 2022.

Spiroiu, I., Erhan, C., Crișan, R.-D.-N., Avramiuc, N., and Flueraș, M.: Geo-gravimetric quasi-geoid determination over Romania, *Journal of Geodesy, Cartography and Cadastre*, 2017.

The National Center for Cartography: Determination of a quasigeoid for the area of Romania: https://www.cngcft.ro/index.php/ro/portofoliu/proiecte-in-lucru/determinarea-unui-cvasigeoid-pentru-zona-romaniei, last access: January 15, 2022.

The National Center for Cartography: Carrying out leveling measurements, GNSS and oceanographic determinations at tidal waves in ports of Constanța, Mangalia, Sulina for tracking the oscillations of the Black Sea level compared to the altitude "0": https://www.cngcft.ro/index.php/ro/portofoliu/proiecte-in-lucru/efectuarea-de-masuratori-de-nivelment-determinari-gnss-si-oceanografice-pentru-urmarirea-in-timp-a-oscilatiilor-nivelului-marii-negre-fata-de-altitudinea-0, last access: January 15, 2022.

The National Center for Cartography: Softwares: https://rompos.ro/index.php/download/category/2-software, last access: January 15, 2022.

Tiberiu, R., Constantin, M., and Danciu, V.: Considerations on the state of Romanian national geodetic network, *RevCAD*, 14/2013, 2013.

Vanicek, P., Kingdon, R., and Santos, M.: Geoid versus quasigeoid: a case of physics versus geometry, *Contributions to Geophysics and Geodesy*, Vol. 42/1, 2012.