DOI: 10.1051/bioconf/20160701035

Methodology and software for georeferencing vineyards

Flávio Bello Fialho

Researcher, Embrapa Uva e Vinho, Rua Livramento 515, Bento Gonçalves, RS, Brazil

Abstract. An agricultural registry is a collection of information about production area and yield of agricultural properties in a region or designated area. It allows to measure agricultural production and its spatial distribution, characterize rural structure, facilitate inspection and development of agricultural policies, optimize distribution of agricultural credit, estimate crop yield and generate research data. A key component for a quality registry is accurate measurement of areas and their geographical position, through georeferencing, to allow integration with other spatial information. The Vineyard Registry of Rio Grande do Sul is one of the most complete agricultural registries in Brazil. It has been carried out in all grape producing properties in the state since 1995, and its georeferencing began in 2005, with the objective of accurately map vineyards. Embrapa has developed a methodology to accelerate georeferencing, by simplifying the field mapping process. One of the central points of this methodology was the development of a software called MapaGPS to organize and classify measured points in the field. Recently, this software has been improved, with the incorporation of features, such as transformation between coordinate systems, conversion between files of different formats, and more control over generated charts. The georeferencing experience of the Vineyard Registry of Rio Grande do Sul may be used throughout Brazil and other countries. The software is available under a free license, and there are no restrictions to adopting the methodology. This document aims to disclose details of this methodology and how it may be used to facilitate zoning projects worldwide.

1. Introduction

An agricultural registry is a collection of information about production area and yield of agricultural properties in a region or designated area. Agricultural registries serve many purposes. Tthey may be used to measure agricultural production and its spatial distribution, or to characterize the social and economic structure of rural areas. An agricultural registry may be used to verify crop insurance claims, or for traceability purposes. Government agencies may use it in the development of agricultural policies, to optimize distribution of agricultural credit or to estimate crop yield. Research institutes may use it to gather collective data for spacial analysis.

The Vineyard Registry of Rio Grande do Sul is one of the most complete agricultural registries in Brazil. It has been carried out in all grape producing properties in the state since 1995. It is a legal condition that all the grapes commercialized in Rio Grande do Sul must be declared annually in the Vineyard Registry.

A key component for a quality registry is the accurate measurement of areas and their geographical position. By georeferencing the vineyards, ordinary data may be organized as spatial information and subjected to spatial analysis, greatly enhancing the quality of the knowledge derived therefrom. Georeferencing the Vineyard Registry of Rio Grande do Sul began in 2005 [1], with the objective of accurately mapping vineyards. Since then, over one third of the state's vineyards have been mapped, including most of those located within the areas of the currently existing geographical indications for fine wines.

Ideally, it would be possible to trace vineyards over aerial images using Geographical Information System (GIS) software. However, in many situations, it is difficult to distinguish sectors within a vineyard, as may be seen in Fig. 1. This makes it important, albeit expensive and time consuming, to collect data on the field, using GPS receptors. In order to optimize resources, a methodology for collecting geographical data and a software to process data and trace the vineyards and its subdivisions were developed by Embrapa [1]. The software, called MapaGPS, has been recently rewritten to include many new features, and is freely available for download at *http://sistemas.cnpuv.embrapa.br/mapagps*, for free use and distribution under the GPL license [2].

This work describes the georeferencing process used and how to use the software MapaGPS to map vineyards. Hopefully, this experience may be useful for georeferencing vineyards, as well as other crops, in Brazil or other parts of the world.

2. Method of measuring data points

The Vineyard Registry organizes the grape producing area into properties, each with an individual *registry code*. Within a property, the grape producing area is divided into vineyards and sectors. A *vineyard* is defined as a continuous area where grapes are grown. If two areas are not connected, they are considered separate vineyards. Vineyards within a property are identified by integer *vineyard numbers*. Each vineyard may be divided into sectors, where a *sector* is a continuous area with the same characteristics (cultivar, age, conduction system, spacing, and so forth). Sectors within a vineyard are identified by a single number or letter (1 . . . 9, a . . . z) called the *sector code*. The "special" sector '0' (zero) is reserved

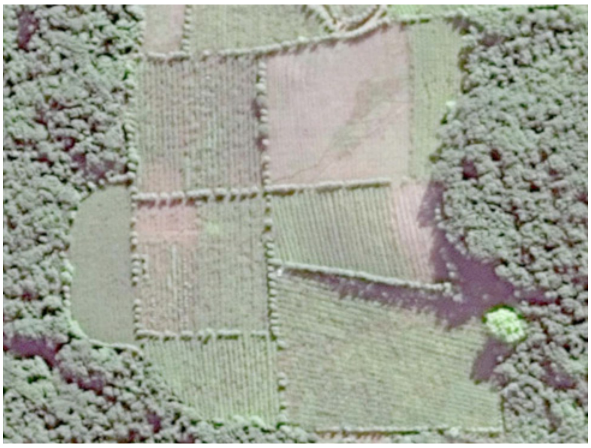

Figure 1. Aerial image of a vineyard to be georeferenced.

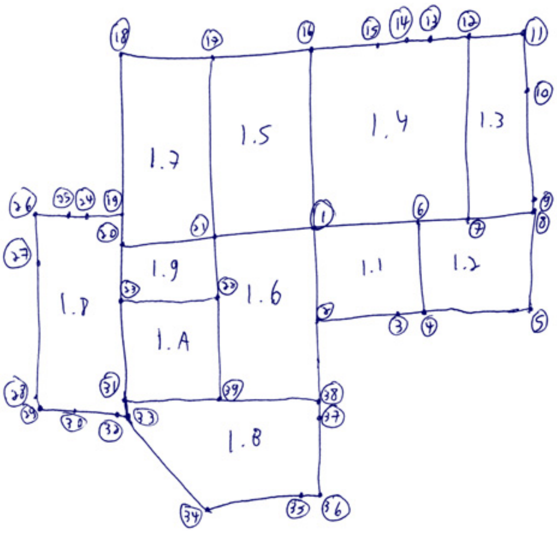

Figure 2. Hand-made sketch of a vineyard, showing sector labels and point IDs.

for areas where grapes are not grown. Sector are uniquely identified within a property by a *sector label*, formed by the vineyard number followed by a period and the sector code (e.g. 1.1, 1.3, 2.7, 2.b, 15.1).

The first step to take when measuring a property in the field is to draw a rough sketch of the vineyards and their sectors on a sheet of paper and plan the order in which points will be measured. This sketch may look something like Fig. 2. When contrasted with Fig. 1, it becomes evident that, while some divisions between sectors are clear in the aerial image, others are not. The sketch on Fig. 2, after all measurements and processing are done, may be accurately represented by the plot on Fig. 3.

A key point in the method is to properly identify the points being measured. Each *point name* is formed by the vineyard number followed by a period and the sector codes of all sectors which are in contact with the point, including sector '0' (in case the point is in contact with the external perimeter, or with an internal area without vines). For example, the correct nomenclature for point 1 in Figs. 2 and 3 is '1.1456', because this is the point where the lines which delimit sectors 1, 4, 5 and 6 meet. Similarly, point 2 is named '1.016', because it touches the area outside the vineyard (sector '0') and sectors 1 and 6, and point 3 is '1.01', because it is in the middle of the line which divides sector 1 and the area outside the vineyard (sector '0'). Point names are not unique, and more than one point may have the same name.

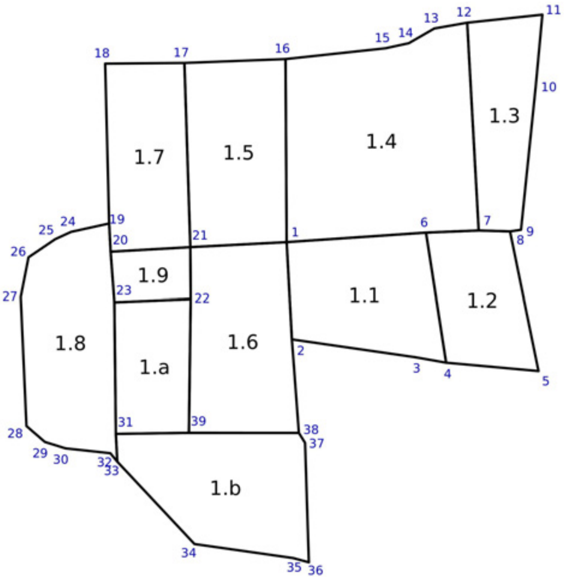

Figure 3. Accurate plot of a vineyard, showing sector labels and point IDs.

An important concept in the method is that data points may be classified as "line points" or "corner points". *Line points* are those which join only two line segments, and consequently are in contact with only two sectors. One example of a line point in Fig. 3 is point number 3, which is only in contact with sector '1' and the external sector '0', thus being labeled '1.01'. Other examples of line points are points 5, 9, 10 and 11. *Corner points*, on the other hand, join three or more line segments and are in contact with three or more sectors. One example of a corner point in Fig. 3 is point number 2, which is in contact with sectors '1', '6' and the external sector '0', thus being labeled '1.016'. Other examples of corner points are points 1, 4, 6 and 7.

Point coordinates should be measured using a precision GPS receiver, and the measurement error, after data post-processing, should be no more than 50 cm. After all points are measured, the next step is to download the data from the GPS receiver to a computer, apply any necessary corrections and store data points, one per line, into a text file in the format shown in Fig. 4. Each line of this file corresponds to a measured point, with whitespace-separated fields containing the following data: latitude (decimal degrees), longitude (decimal degrees), point name (as described earlier), date of measurement (in the format YYYY-MM-DD), time of measurement (in the format HH:MM:SS), altitude (meters), UTM Y coordinate (meters), UTM X coordinate (meters) and *point ID* (an integer which describes the order of measurement of the point within the property).

There are two restrictions on the order in which points should be listed in the data file. First, line points which form a continuous line (and therefore have the same label) in the sketch *must* be listed in sequence. For the vineyard in Fig. 3, points 9, 10 and 11 (all of which

```
-29.163718409  -51.527214523  1.012  2015-06-30  18:00:01  631.732  6773760.616  448730.085  1
-29.163424942  -51.527195996  1.02   2015-06-30  18:05:02  627.039  6773793.140  448731.741  2
-29.163456667  -51.526384018  1.02   2015-06-30  18:10:03  628.778  6773789.978  448810.720  3
-29.163738489  -51.526722728  1.012  2015-06-30  18:15:04  634.555  6773758.606  448777.921  4
-29.164069724  -51.526891317  1.01   2015-06-30  18:20:05  637.903  6773721.833  448761.690  5
-29.164062878  -51.527226720  1.01   2015-06-30  18:25:06  634.929  6773722.445  448729.070  6
-29.164059924  -51.526758853  2.01   2015-06-30  18:30:07  637.001  6773722.976  448774.567  7
-29.163605797  -51.526384534  2.01   2015-06-30  18:35:08  633.577  6773773.455  448810.744  8
-29.163628429  -51.525890371  2.01   2015-06-30  18:40:09  634.945  6773771.162  448858.812  9
-29.164104159  -51.525900520  2.01   2015-06-30  18:45:10  641.266  6773718.449  448858.060  10
```

Figure 4. Example of a MapaGPS input file, with one data point per line.

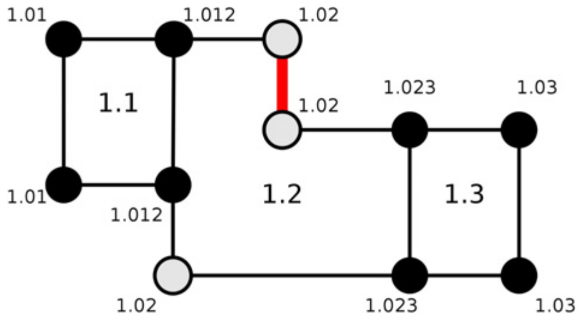

Figure 5. Example of line point restrictions. The two line points labeled '1.02' at the top must be listed together. Both points must be separated from the point labeled '1.02' at the bottom by at least one other point.

should be labeled '1.03') must be listed in sequence. The sequence may be reversed, so {11-10-9}is as acceptable as {9-10-11}. The other line sequences which must be preserved in order, in this example, are {131415}, {24-25-26-27-28-29-30-32}, and {34-35-36-37}.

The second restriction is that, if two or more sequences of line points with the same label are not connected in the sketch, they *must not* be listed in sequence in the file. The vineyard shown on Fig. 5 has two sequences of line points labeled '1.02', the first formed by the two points on the top of sector 1.2 and the second formed by the single point on the opposite side of the sector. In the data file, the two points of the first sequence must come one immediately after the other, and they must be separated from the third '1.02' point by at least one of the other points in the vineyard.

3. Software for joining data points

In order to help the process of georeferencing the Vineyard Registry of Rio Grande do Sul, a software called MapaGPS was developed by Embrapa [1,2]. This software is a command-line utility which reads data files in the format previously specified, orders the points of each sector and writes output files in a text format, which may be easily read into the Vineyard Registry. The output files list all sectors of the vineyard and their coordinates, in order. A PDF file with an accurate plot of the vineyards and sectors of each property is also produced, and other output formats may be generated as well.

The program was made available for download as free software at *http://sistemas.cnpuv.embrapa.br/mapagps*. Both source code and precompiled binaries for Linux 32 bits (*mapagps-i686.tar.gz*) and Linux 64 bits (*mapagps-x86_64.tar.gz*) are available. To install the binaries, simply extract the files in the archive and move the file *mapagps* (the binary executable) to a directory in the *$PATH*. The program uses libraries *libcairo2*, *libmagic1*, *libproj9*, *libshp2* and *libxml2*, which should be installed previously, if not already present.

For other platforms, the program may be easily compiled from the source code, using a standard C language compiler, such as gcc. After installation, the program may be executed using the command '*mapagps*', followed by options and the names of the input files. Help is available by typing the command followed by the '—h' option, which will show the expected syntax and briefly describe all available options:

```
mapagps -h
```

For the most common case, given the input file *RS00000.txt*, the command:

```
mapagps RS00000.txt
```

will read the input file and generate a PDF file with the plot of the vineyards and an output file named '*RS00000.out*' in the format shown in Fig. 6.

The output file format is self-explanatory, when compared to the input file. The registry code and the total vineyard area of the property is shown on the first line. Next, each vineyard and its sectors are listed, one per line, with their respective areas and the list of ordered points of the perimeter of each sector (one point per line). Compared to the lines in the input file, each point is preceded by one additional field containing the sector code. The perimeter of the vineyards is described by the points of sector '0' of each vineyard. Lines identifying the registry code, vineyard number and sector label, and points of sector '0' are preceded by a '#' character.

The program accepts input formats other than the one shown on Fig. 4. It is able to read its own output files (like the one shown on Fig. 6), ESRI Shapefiles [3] and XML files compatible with the specifications of the OpenStreetMap (OSM) project [4]. In each of these three cases, registry codes are stored within the files, and a single file may contain more than one property.

ESRI Shapefiles [3] are commonly used when exchanging data with GIS software. Shapefiles are composed of three files with extensions *'.shp'*, *'.shx'* and *'.dbf'*, and should follow a specific format. Sectors are represented by entities, which should be of type *POLYGONZ*, with X, Y, Z and M components of each point containing longitude (in decimal degrees), latitude (in decimal degrees), altitude (in meters) and point ID (an integer). The program also accepts Shapefiles with entities of types *POLYGONM* (without the unique point ID) or *POLYGON* (without altitude and point ID). UTM coordinates, in meters, may also be used in the X and Y

```
## CADASTRO RS00000 (0.7046 ha) ##

# VINHEDO 1 (0.3589 ha)
# Setor 1.0 (0.3589 ha)
# 1.0  -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1
# 1.0  -29.163424942 -51.527195996  1.02   2015-06-30 18:05:02  627.039  6773793.140 448731.741  2
# 1.0  -29.163456667 -51.526384018  1.02   2015-06-30 18:10:03  628.778  6773789.978 448810.720  3
# 1.0  -29.163738489 -51.526722728  1.012  2015-06-30 18:15:04  634.555  6773758.606 448777.921  4
# 1.0  -29.164069724 -51.526891317  1.01   2015-06-30 18:20:05  637.903  6773721.833 448761.690  5
# 1.0  -29.164062878 -51.527226720  1.01   2015-06-30 18:25:06  634.929  6773722.445 448729.070  6
# 1.0  -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1
# Setor 1.1 (0.1519 ha)
1.1    -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1
1.1    -29.163738489 -51.526722728  1.012  2015-06-30 18:15:04  634.555  6773758.606 448777.921  4
1.1    -29.164069724 -51.526891317  1.01   2015-06-30 18:20:05  637.903  6773721.833 448761.690  5
1.1    -29.164062878 -51.527226720  1.01   2015-06-30 18:25:06  634.929  6773722.445 448729.070  6
1.1    -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1
# Setor 1.2 (0.2070 ha)
1.2    -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1
1.2    -29.163424942 -51.527195996  1.02   2015-06-30 18:05:02  627.039  6773793.140 448731.741  2
1.2    -29.163456667 -51.526384018  1.02   2015-06-30 18:10:03  628.778  6773789.978 448810.720  3
1.2    -29.163738489 -51.526722728  1.012  2015-06-30 18:15:04  634.555  6773758.606 448777.921  4
1.2    -29.163718409 -51.527214523  1.012  2015-06-30 18:00:01  631.732  6773760.616 448730.085  1

# VINHEDO 2 (0.3457 ha)
# Setor 2.0 (0.3457 ha)
# 2.0  -29.164059924 -51.526758853  2.01   2015-06-30 18:30:07  637.001  6773722.976 448774.567  7
# 2.0  -29.163605797 -51.526384534  2.01   2015-06-30 18:35:08  633.577  6773773.455 448810.744  8
# 2.0  -29.163628429 -51.525890371  2.01   2015-06-30 18:40:09  634.945  6773771.162 448858.812  9
# 2.0  -29.164104159 -51.525900520  2.01   2015-06-30 18:45:10  641.266  6773718.449 448858.060  10
# 2.0  -29.164059924 -51.526758853  2.01   2015-06-30 18:30:07  637.001  6773722.976 448774.567  7
# Setor 2.1 (0.3457 ha)
2.1    -29.164059924 -51.526758853  2.01   2015-06-30 18:30:07  637.001  6773722.976 448774.567  7
2.1    -29.163605797 -51.526384534  2.01   2015-06-30 18:35:08  633.577  6773773.455 448810.744  8
2.1    -29.163628429 -51.525890371  2.01   2015-06-30 18:40:09  634.945  6773771.162 448858.812  9
2.1    -29.164104159 -51.525900520  2.01   2015-06-30 18:45:10  641.266  6773718.449 448858.060  10
2.1    -29.164059924 -51.526758853  2.01   2015-06-30 18:30:07  637.001  6773722.976 448774.567  7
```

Figure 6. Example of a MapaGPS output file, with the ordered points of the perimeter of each sector.

components of the Shapefile, if option '−u' is specified in the command line. The associated database should have four fiels: a string field named *'cadastro'* containing the registry code, an integer or string field names *'vinhedo'* containing the vineyard number, a single-character string field named *'setor'* containing the sector code, and a string field named *'datahora'* containing the date and time of measurement of data points in the sector, in the format 'YYYYMMDD HH:MM:SS'.

OpenStreetMap [4] is a project to create a free map of the world. It exchanges data using an XML file format, using filkes with extension *'.osm'*, which may be read by MapaGPS. Within the XML file, vineyards are recognized by use of the tag *'landuse=vineyard'*, which may be applied to a closed way or to a multipolygon relation. Vineyards should also be identified by the use of the tags *'cadastro'*, *'vinhedo'* and *'setor'*, containing the registry code, vineyard number and sector code, respectively. Data points may have the tags *'n_ord'*, *'cod_ponto'* and *'ele'*, representing point ID, point name (following the standards described earlier) and elevation (altitude, in meters). If relations are used, the ways within each relation must be in order.

MapaGPS normally generates an output file in plain text format with the extension *'.out'*. If the file exists, it is not overwritten, but a file with extension *'.out.novo'* is generated instead. It also generates a single-page PDF file with a plot of the vineyards. The names of these files are given by the registry code, which is read from the input files or derived from the input file name. The program will also generate Shapefiles, if the option '−s' is given, and OSM XML files, if the option '-o' is given. It is also able to generate a file in the input format, with the extension *'.txt.novo'*, by using the '−t' option. Each of these generate one file per registry code. If more than one property is being processed simultaneously, it is possible to generate PDF, output, Shapefile and OSM files containing all properties processed in a single file of each type, by using the '−m' option. The global PDF will contain multiple pages, one per registry code.

Options may be combined. For example, if all '.txt' files in a directory contain information about vineyards measured in the format shown on Fig. 4, the options '−s', '−o', '−t' and '−m' may be used together to generate all possible formats of file output with the command:

```
mapagps -s -o -t -m *.txt
```

This will generate one file of each type for each registry code encountered, and will also generate the global Shapefile *mapagps* (composed of the three files *mapagps.shp*, *mapagps.shx* and *mapagps.dbf*) and the global files *mapagps.out*, *mapagps.pdf* and *mapagps.osm*. If these files exist, they are overwritten, except for *mapagps.out*, which will have the new text appended at the end of the file.

It is also possible to suppress writing individual PDF, output, Shapefile and OSM files, using option '−n'. When used by itself, this option avoids writing files altogether, which may be useful if one only needs to check the input files for errors. This option may also be used in conjunction with the '−m' option, and possibly '−s' or '−o', in order to only generate the global files, but not he individual ones. It does not, however, cancel the '−t' option, and both may be used together to generate only the new input files.

MapaGPS usually does not print messages on-screen, unless problems with the data cause it to print warnings or error messages. Option '−v' may be used in order to increase verbosity and show more information. If used multiple times, more detailed messages are shown. On the other hand, option '−q' my be used to suppress warnings. If used twice, errors will not be shown either.

As seen in Fig. 4, the input file contains both geographical (latitude and longitude) and planar (UTM) coordinates. MapaGPS normally uses geographical coordinates, unless the values of latitude and longitude are zero. It compares them with the planar coordinates and, depending on the verbosity level, issues a warning if they are off by more than about 50 cm. However, if the option '−u' is used, planar coordinates are preferred. Option '−l' may be used to change back to geographical coordinates.

The official reference system used in Brazil is SIRGAS2000 [5], which is equivalent to WGS84, and it is the program's default. The UTM zone is calculated using the geographical coordinates of the first point read by the program, unless they are zero, in which case it defaults to zone 22 south (the zone where most brazilian vineyards are located).

It is possible to read data in a different coordinate system by specifying PROJ.4 parameters (easily available on the Internet) in the '−u' or '−l' options. For example, the command:

```
mapagps -l"+proj=longlat +ellps=aust_SA
+towgs84=-67.35,3.88,-38.22
+nodefs" RS00000.txt
```

processes file RS00000.txt using the SAD69 datum and geographical coordinates. Note that the command should be typed in a single line and that there must be no space before the starting quotes. Similarly, in order to use the SAD69 datum and UTM zone 21 south coordinates, one would type (in a single line):

```
mapagps -u"+proj=utm +ellps=aust_SA
+towgs84=-67.35,3.88,-38.22 +zone=21
+south +nodefs" RS00000.txt
```

The plot stored in the PDF file may be controlled by parameters '−g' and '−e'. Text size and line thickness is multiplied by the value of parameter '−e'. For example, if '−e2' is used, text will be twice as large and lines twice as thick. On the other hand, if '−e0.5' is used, they will be half the usual size (thinner lines and smaller text). It is also possible to control which elements will be shown on the graph by the value of parameter '−g', which should be the sum of the values of thew desired elements:

- 1: X and Y axes and scales
- 2: Graph title (registry code and total area)
- 4: Sector perimeters, in black
- 8. Sector codes
- 16: Vineyard perimeters, in green (red on error)
- 32: Point IDs, in blue

The default value is '−g15', if there were no errors, or '−g63', if an error occurred in one of the vineyards. In this case, the perimeter of the vineyards in error will be in red instead of green, and the graph title will also be in red. An example of a vineyard with an error is shown on Fig. 7, in which the program was not able to correctly join the data points of vineyard 1, although vineyard 2 is fine.

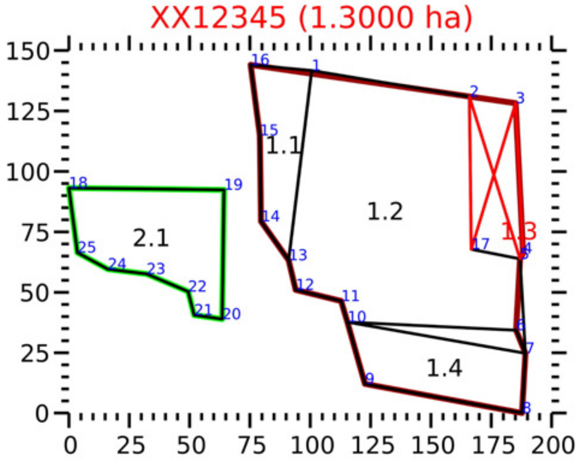

Figure 7. Plot of vineyards showing all graphical elements (option '−g63') and the visual representation of errors while processing vineyard 1.

In the vast majority of cases, MapaGPS is able to read and process data points without problems. However, sometimes errors do occur, and the input data file must be edited to correct them. Common syntax errors include the use of a decimal comma instead of a decimal period, accidentally removing a minus sign in front of a geographical coordinate, or incorrect, extra or missing fields. Point nomenclature is very strict, and any incorrect point names must be fixed. Another common problem is not following the restrictions on line points explained earlier. All these problems are very easy to identify and fix.

More rarely, the program is not able to correctly order data points in one or more sectors due to limitations of the algorithm. In these cases, the order of the data points must be changed in the input file, until the program is able to execute correctly. One possibility which usually works is to reorder the points of the troublesome vineyards by starting at a corner point and listing the points in an order which follows the vineyard's perimeter. At the end, the internal points should be listed. In the rare cases where this does not work, it may be necessary to move points around using trial and error, until an adequate order is found.

It is also possible that the program is able to join the points, but the generated plot does not agree with the field sketch. In these cases, the same procedure described to deal with errors may be applied.

In most cases, however, the program executes with no errors. After processing, the output text file created by the program is read into the Viticultural Registry database, along with the generated plot of the vineyards, which is stored for future reference.

4. Test of the program

In order to verify if the use of MapaGPS was practical or not, a large number of properties measured on the field with GPS receptors was processed, without any previous changes in the raw data. The objective of this test was to estimate how well the program could trace the vineyards without human intervention, if the methodology were followed correctly.

The most recent version of MapaGPS was tested on data from 2761 grape producing properties. Of these, 2516 (91.1%) were processed without error on the first run of the program, 20 (0.7%) only had minor syntax errors which required small changes in the file, and 132 (4.8%) had not followed the line point restrictions mentioned earlier. Fixing these methodological errors solved these problems, leaving the files in the form that they should have arrived.

In the remaining properties, the order of points of the troublesome vineyards was rearranged as suggested earlier, and that was enough to correctly process 76 (2.8%) cases. The remaining 17 (0.6%) cases had to go through the more troublesome trial and error procedure, but all were eventually processed by the program.

Is is interesting to note that most of the problem data came from properties with at least one vineyard or sector composed of more than one polygon (they either had a "hole" or there was a vineyard made of two sectors joined by a single point). Although this situation only occurred in 67 cases (2.4% of the total), 54 of these required reordering of data points.

The overall success rate of the program, after correction of simple mistakes was 96.6% of all properties, and was considered satisfactory. However, when considering only cases with one polygon per sector, the success rate reaches 98.6%, while the success rate on cases with more than one polygon per sector falls to 19.4%. Therefore, it is advisable that the data be examined more thoroughly, when there are sectors with more than one polygon.

5. Conclusions

The methodology developed and used in the process of georeferencing the Viticultural Registry of Rio Grande do Sul is adequate. It provides a solution for the problem of delimiting sectors within a vineyard which are visually similar in an aerial image.

The MapaGPS software is able to efficiently determine the limits and divisions of vineyards, with a measured success rate of 96.6% on the test cases. It is able to read and write files in a variety of formats and may be used to integrate data from different sources (GPS receivers and data derived from aerial image processing) into the Viticultural Registry. Additionally, the program generates a plot of the vineyards, which may be used by the grape grower for planning and record keeping, and is able to convert data into formats used by GIS software.

The methodology and software may be freely used for georeferencing agricultural registries worldwide. The software source code and executables are freely available, and may be modified to suit individual needs, including translation to other languages. This makes it suitable for use in agricultural registries or other mapping projects in any part of the world.

References

[1] F. B. Fialho, L. M. R. de Mello; L. C. Guzzo, *Metodologia de georreferenciamento do Cadastro Vitícola*, Documentos 50 (Embrapa Uva e Vinho, Bento Gonçalves, 2005)

[2] F. B. Fialho, *MapaGPS 2.0: Software para Georreferenciamento do Cadastro Vitícola*, Circular Técnica 131 (Embrapa Uva e Vinho, Bento Gonçalves, 2016)

[3] ESRI, *ESRI Shapefile Technical Description: An ESRI Whitepaper* (1998)

[4] OpenStreetMap, `http://www.opestreetmap.org` (2016)

[5] IBGE, *Resolução do Presidente do IBGE n° 1/2005* (2005)