Mapping the Via Appia in 3D

Project characteristics: Data management and visualization

O. Martinez-Rubi
Netherlands eScience Center,
Science Park 140 (Matrix 1), 1098 XG Amsterdam, the Netherlands

November 8, 2013

1 Introduction

This project aims to build a 3D Geographic Information System (3D GIS) for the management, visualization and interaction of the 3D spatial data of the Via Appia. This report contains a description of several characteristics of the system, including its input data, its requirements and needs, and its challenges. In addition it contains some suggestions on possible approaches or ways to go. All the information of this report is derived from the visit to the Via Appia in Roma and the conversations with Rens de Hond and Dr. Jeremia Pelgrom.

2 Input data

This system has three different input data types:

1. **LIDAR LAS files**. There are 2 km, along the Via Appia, of PointCloud (PC) data collected by Fugro DriveMap. The estimated size of all the LAS files is 20 GB.

2. **Images of the different 400 monuments**. For each monument or object, which has a *ObjectId*, a LAS file can be exported which contains a PC that combines all the images of the monument. For future reference we call these files Image LAS files and their point clouds Image PC. They have much better resolution than the LIDAR LAS files but they are not aligned with them, i.e. they have different reference system (a local one). The estimated size of these Image LAS files is 60 GB. In section 5.1 there is an extended discussion on how to align these data.

3. **An attribute-data DB**. This Microsoft Access database contains attributes related to the monuments/objects found in the Via Appia and it uses the same *ObjectId* used for the Images. One example of these attributes are the GPS coordinates of the footprint of the monument/object. Another example is a field that indicates if certain part of the monument is marble. Currently this DB is empty, they have some data in files and paper but it still needs to be inserted in the DB.

3 General information

In this section there is a description of several items that must be taking into account for the design of the system.

- The LIDAR PC is basically used to get a general picture of the Via Appia and for its visualization, the more detailed images of its monuments will be given by the Image PC.
- There are only 400 monuments so manual processing on them is an option, though not ideal.
- There will be a maximum of 3 simultaneous users querying data of the developed system.
- The total input data size is expected not to exceed 100 GB.
- The most common query to the system will be selecting points in and around a monument.

4 Requirements and needs

- Possibly the most important requirement (and challenge) that the system has to fulfil is that it has to allow the users to add/modify attributes to a selection of points related to a monument (or parts of a monument). These attributes are not only the ones from the already existing attribute DB but also other ones that may be required in the future. In section 5.2 there is a discussion on possible ways of doing this.
- In the future they will like to filter out objects (like the trees or the trash bins for example). Some ways of doing it:
  - One of the attributes could be if the points can be ignored.
  - New point clouds are created where not-desired points are cleaned out.
- They will make reconstructions of the monuments (which will be meshes) that should be stored in the developed system. These reconstructions could be multiple as a function of time. It needs to be investigated which is the best way of storing them.
- A tool to extract the attribute data from the Microsoft Access DB and assign it to the points should be developed.
- They do not have any hardware to run the system. Can they have use some Cloud system or some external server?

Visualization requirements and needs:

- They want to visualize meshes (not PC).
- Now they have problems when loading few 3D meshes. Maybe we could help by running part of visualization at the server side. Is web visualization an option?
- During visualization they want to be able to zoom in/out. How to use LoD (Level of Detail) with PC should be investigated.
- Interactive visualization, for example move a brick that is on the ground to a location on the monument surface.
- They want to be able to select whether they visualize all the points or only the ones from the Image PC or only the ones from the LIDAR PC. In addition they want to be able to plot the possible reconstructions of monuments (which may also depend on time) instead of (or together with) the points.

5 Proposed approach

The proposed approach for the Via Appia data management is to use a spatial DB to store the points, and more particularly PointCloud DB technologies (like Oracle Spatial PC or Postgres PC). Using these technologies we would group the points in blocks (they are spatially grouped, points closer in space are most likely in the same block). Given the two different input PCs, it seems a good idea to have the data split in the DB as well, i.e. by having two tables: one for the LIDAR PC and another one for the Images PC. In this case we could use 2 processors for the data querying.

An additional requirement that has been discussed related to the proposed approach is that any blocking technique in the point cloud storage must be transparent to the final user but, in principle, this is the case.

Regarding the data loading, the current images of the monuments can be easily converted to LAS files but these need to be aligned with the LIDAR LAS before loading the Image LAS files in the DB. See section 5.1 for more information on how to align the Image LAS files.

For adding and modifying attributes in the DB system we would use a third table which would contain polygons which are bounding boxes of selection of points and we would assign the attributes to these polygons. However, this is not the only proposed approach to deal with adding and modifying attributes in the DB system, see section 5.2 for more information about this approach and the other ones.

A suggestion regarding how to store the reconstructions of the monuments is to directly store the 3D polygons that form them in an additional table in a similar way the attributes would be stored.

5.1 Aligning Image PC with LIDAR PC

Apparently the alignment can be done with the Cloud Compare SW. The steps are:

- Load a reference LAS file from the LIDAR PC. This LAS File must contain the monument.
- Load the Image LAS file of the monument.
- Align the Image PC data by defining 3 points that should match with the LIDAR PC.
- Once aligned, save the new LAS file with the aligned Image PC data (only this one, we do not want to add to the new LAS file the data from the LIDAR PC). Our hypothesis is that in this new LAS file the XYZ would be in the correct reference system. This still needs to be checked.

Loading a LIDAR LAS file in Cloud Compare can be too heavy if the file is large. A suggestion is to only load the points that we know will overlap with the Image LAS. For this we can first load all the LIDAR LAS in the PC (in its own table) and use queries to get the points that correspond to the positions of the several objects, then save LAS files of these points to be used as alignment for the Image LAS file. We could automate the reference LAS file generation since we have GPS positions of the footprint of the several monuments (in the attribute data).

Once Image LAS files are aligned they can be loaded in the DB PC (in a different table than the LIDAR PC). They do not need to be loaded all at a time. We can start loading few of them as they become available

It would be good to keep a reference between the *ObjectId* and the points in the DB (for example: maybe we need to delete all the points related to a monument from the DB if we realize they are not properly aligned)

5.2 Adding (and modifying) attributes

The adding and modifying of the attribute data in PC appears to be not a trivial task. Here you can find three suggestions that should be tested to see which one adapts better to the projects needs.

1. Use PC and store the modifiable attributes only in the Image PC (so, LIDAR PC is static and Image PC is "updateable"). In this case the updating is not very elegant since updating attributes in a point means get the block (where the point is), unpack it (and uncompress it), modify the attribute, repack the block (and compress it) and replace old block with the new one. Also from time to time we would have to vacuum the database and maybe re-index it. In addition it is not clear how to add new columns (attributes) to the PC, this should be investigated. See section 5.2.1 for an idea on how to implement updating attributes in PC.

2. Use PC but neither LIDAR PC nor Images PC would be updated. We would store all the attributes (that can be modified) into separate table. Actually, in this table we would define 3D polygons (bounding boxes of the selected points) and assign attributes to these polygons. In the table only the polygon (which, again, is the bounding box of the selected points) and the attribute is stored. Note that, in this case, access to the points within the polygon will require queries to the PC tables. Also note that we would have 3 tables, 2 PC and the polygons table with attributes.

3. Another option is to forget using PC technologies and store each point with all the attributes in a row (still using PostGIS or other spatial extension of course). In this case the points are easy to access/modify. Depending on the total amount of points this idea is maybe interesting.

After initial consideration it seems that the best option would be the 2 because when using the attribute data it seems to be enough with just using the bounding boxes (not the actual points) and this decreases considerably the amount of data to be stored. Still the recommendation is to try all the approaches and finally use the best one. In the end the data size is not too large so it should not be a problem. Also, given the size, making the tests should not take too long.

5.2.1 Updating attributes in PC

After checking with the Postgres PC users forum it seems that, right now, modifying attributes in PC using blocks means, indeed, get the block, unpack it (and uncompress it), modify the points, repack the block (and compress it) and replace the old block with the new one. This still reinforces more the selection of the commented approach, i.e. store only the bounding boxes or 3D polygons of the points in which we want to add/modify the attributes, in this way assigning such attributes to only these polygons.

However, here we present how the update of attributes could be done if we want to store them in the blocks, i.e. to a point level:

1. The user would select some area where he wants to add an attribute. This area contains several blocks.

2. Create a candidate table that contains all blocks that overlap with the selected region. These blocks contains points that may actually be out of the selected area.

3. Create a results table which only contains the points that really are in the selected area. These points have been unpacked from the blocks. The points should still contain the information of to which block they belong.

4. Create table with the excluded points.

5. The modification of the attributes by the user will mean that some values are modified in the results table.

6. After modification, we can create back the blocks (in fact the modified versions of them) by combining both results and excluded tables. Afterwards, the old blocks are replaced with the new ones with just an *UPDATE* operation.

5.3 Next recommended steps

As already stated, it is crucial to make some tests to see which option to handle the attributes data fits better in this project. Assuming that the test would reveal that best option is, indeed, the approach 2 described in section 5.2, the next steps to follow related to the data management would be:

1. Load LIDAR LAS files in PC DB (using PC techniques) in table 1. table 2 will contain the data from aligned Image LAS files (but we still need to align them) .

2. For each of the 400 objects:

   - Get the LIDAR PC points that are within its footprint and create the reference LAS file.
   - Load reference LAS file in Cloud Compare SW.
   - Load the Image LAS file from the object/monument.
   - Align the Image PC data in the Cloud Compare and store it (only the Image PC data, i.e. not the LIDAR PC) as a new Image LAS file.

3. When new aligned LAS images files (related to the several monuments) become available, they can be loaded in table 2.

4. Start thinking what attributes will be in the polygon table 3.