Analysis of available tools

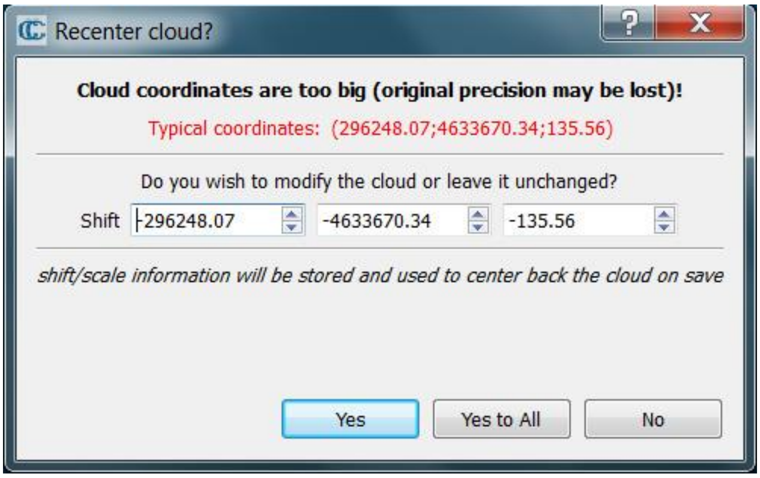
WP1 deliverable – Mapping the Via Appia in 3D – NLeSC & SPINlab VU University Amsterdam – by Rens de Hond

This document provides an evaluation of existing software and tools that could be of use for the development of a 3D GIS as described in the "User requirements" document. Which tools are already available and can be used, which functionalities need to be developed, and are those programs or tools extensible or re-useable?

CloudCompare[1]

CloudCompare is free and open-source. "CloudCompare is a 3D point cloud (and triangular mesh) processing software. It has been originally designed to perform comparison between two 3D points clouds (such as the ones obtained with a laser scanner) or between a point cloud and a triangular mesh. It relies on a specific octree structure that enables great performances in this particular function. It was also meant to deal with huge point clouds. Afterwards, it has been extended to a more generic point cloud processing software, including many advanced algorithms."[2]

- CloudCompare contains a 3D viewer with basic 3D functionalities for navigating.
- CloudCompare has an **alignment tool** for aligning and scaling point clouds and meshes. The alignment is not done automatically though. At least three pairs of equivalent points need to be indicated by hand on both clouds. The result is acceptable for site 162 (pyramid), but it does not work for a smaller site like site 154, which is only a building block and is not covered by so many points in the DRIVE-MAP point cloud.
- When loading a block of DRIVE-MAP .las data (for example Rome-000062.las), CloudCompare wants to recenter the cloud because the coordinates are too big. It says that the shift/scale information will be stored and used to center back the cloud on save. It shows this message:



- CloudCompare might be usable as a **user interface**, since it is has the needed 3D-viewer functionalities, and it is open-source and allows for PointCloudLibrary plugins.
- CloudCompare has a good example of a '**measure in 3D space**' functionality.
- The 'define a rectangular 2D label' functionality might be useful to develop a **segmentation tool**.

[1] http://www.danielgm.net/cc/
[2] http://www.danielgm.net/cc/ > Presentation

- CloudCompare supports the import of OBJ and PLY meshes, useful for the integration of **archaeological reconstruction** models.

MeshLab[3]

MeshLab is free and open-source. "MeshLab is an open-source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes. The system is aimed to help the processing of the typical not-so-small unstructured models arising in 3D scanning, providing a set of tools for editing, cleaning, healing, inspecting, rendering and converting this kind of meshes. The system is heavily based on the VCG library developed at the Visual Computing Lab of ISTI - CNR, for all the core mesh processing tasks."[4]

- MeshLab contains a 3D viewer with basic 3D functionalities for navigating.
- MeshLab has an **alignment tool** for aligning and scaling meshes. It is similar to the one in CloudCompare: at least four pairs of equivalent points need to be indicated manually. MeshLab works with meshes though, and does not support .las files.
- Although MeshLab allows for plugins, it seems to be unsuitable as a base for a user interface, because it based on meshes instead of point clouds.
- MeshLab contains a good example of a '**measure in 3D space**' functionality.
- MeshLab supports OBJ, PLY, Collada, 3ds file formats, useful for the integration of **archaeological reconstruction** models.

PointCloudLibrary[5]

"PCL (Point Cloud Library) is a standalone open-source framework including numerous state of the art algorithms for n-dimensional point clouds and 3D geometry processing"[6]. PointCloudLibrary (PCL) is free and open-source, it is free for commercial and research use.

- Registration library. For combining and **aligning** several point cloud datasets into a global consistent model. *"Combining several datasets into a global consistent model is usually performed using a technique called registration. The key idea is to identify corresponding points between the data sets and find a transformation that minimizes the distance (alignment error) between corresponding points. This process is repeated, since correspondence search is affected by the relative position and orientation of the data sets. Once the alignment errors fall below a given threshold, the registration is said to be complete. The pcl_registration library implements a plethora of point cloud registration algorithms for both organized an unorganized (general purpose) datasets."*[7]
- Filters library. To **filter out trees**? *"Contains outlier and noise removal mechanisms for 3D point cloud data filtering applications. An example of noise removal is presented in the figure below. Due to measurement errors, certain datasets present a large number of shadow points. This complicates the estimation of local point cloud 3D features. Some of these outliers can be filtered by performing a statistical analysis on each point's neighborhood, and trimming those which do not meet a certain criteria. The sparse outlier removal implementation in PCL is based on the computation of the*

[3] http://meshlab.sourceforge.net/
[4] http://meshlab.sourceforge.net/
[5] http://www.pointclouds.org/
[6] http://en.wikipedia.org/wiki/PCL_(Point_Cloud_Library)
[7] http://docs.pointclouds.org/trunk/a02953.html

distribution of point to neighbors distances in the input dataset. For each point, the mean distance from it to all its neighbors is computed. By assuming that the resulted distribution is Gaussian with a mean and a standard deviation, all points whose mean distances are outside an interval defined by the global distances mean and standard deviation can be considered as outliers and trimmed from the dataset."[8]

- Segmentation library. For **segmentatio**n of site point clouds into objects. *"The pcl_segmentation library contains algorithms for segmenting a point cloud into distinct clusters. These algorithms are best suited to processing a point cloud that is composed of a number of spatially isolated regions. In such cases, clustering is often used to break the cloud down into its constituent parts, which can then be processed independently."*[9]
- Visualization library. *"The pcl_visualization library was built for the purpose of being able to quickly* ***prototype and visualize*** *the results of algorithms operating on 3D point cloud data."*[10]

123D Catch[11]

Autodesk 123D Catch "creates 3D models from series of photographs taken at various angles using photogrammetry".[12] It is free but completely closed software. Therefore it is not useable as source for algorithms or as base for the user interface, but is provides some good examples of the intended functionalities.

- 123D Catch contains a 3D viewer with basic 3D functionalities for navigating.
- 123D Catch stores the position and angle of the camera for each photograph that is used for the 123D Catch model. By clicking on the camera position, one flies to that position and can view the 3D model perfectly overlaid by that photograph. This is a good example of we are looking for in the **historical pictures** functionality.
- The GPS data that is stored in the EXIF data of any photograph taken with a GPS camera and used for the 123D Catch models, is *not* usable for **alignment** purposes. The GPS in a camera, which is not a DGPS, simply does not have the necessary accuracy.
- NB: after scaling in 123D Catch, the exported LAS point cloud looks weird in CloudCompare, as if decimated and as if the points are positioned in a raster. So the scale functionality can't be used in the workflow.
- The lasso selection tool is probably a good inspiration for how to select the points that belong to one object in the **segmentation functionality**.
- 123D Catch contains a good example of a '**measure in 3D space**' functionality

Fugro Flaim

Flaim is not (yet) operable on Mapping the Via Appia computers, so an analysis of this software will follow later. Rens is in contact with Fugro to solve this problem.

[8] http://docs.pointclouds.org/trunk/a02945.html
[9] http://docs.pointclouds.org/trunk/a02956.html
[10] http://docs.pointclouds.org/trunk/a02958.html
[11] http://www.123dapp.com/catch
[12] http://en.wikipedia.org/wiki/Autodesk_123D

Fugro Viewer

This point cloud viewer is created by Fugro. It is not allowed to decompile, reverse-engineer or modify the software. It's 3D viewer and tools to create areas of interest might be good examples for a **segmentation tool**.

ArcScene

ArcScene is a 3D visualization environment of the 3D Analyst extension in ESRI ArcGIS, perhaps the most used GIS software in the world. It is closed-source commercial software.

ArcScene 10.1 and 10.2 support lidar data (LAS files).

- ArcScene contains a 3D viewer with basic 3D functionalities for navigating.
- In addition, ArcScene has a very nice "fly" mode for navigating.
- When the entire Via Appia LAS dataset is loaded into ArcScene (nearly 250.000.000 points), the cloud is being decimated, at least for visualisation.
- ArcScene does **not** allow for point cloud alignment and segmentation, and to use segments of the point cloud as GIS features which are connected to a database.
- Apparently, for LAS data from (mobile) terrestrial lidar, which is the kind of data Fugro DRIVE-MAP has collected for the Via Appia, ArcScene should only be used to view the point clouds. In ESRI's own words: "*Many tools for the LAS dataset were designed for airborne lidar where surfaces are commonly interpreted from the point cloud. For example, with airborne lidar it makes sense to toggle on TIN based surface views or to interpolate raster DEMs. Terrestrial lidar, on the other hand, tends to be more immersive and is often only valid to view and use as a point cloud. ArcGIS doesn't know what kind of lidar you have so it's up to you to use it appropriately. For the most part, terrestrial lidar should just be used and viewed in ArcScene as points. The points can be used as a backdrop for digitizing measurements and features and be displayed in an integrated fashion with other GIS layers*."[13]

gvSIG 3D

gvSIG is an open-source GIS platform for which a 3D extension has been developed.[14] The development of this 3D extension has been stopped in 2012.[15] The 3D functionalities are limited to visualisation of simple 3D shapes and extruded 2D features. The required functionalities are lacking, and it only supports .osg .ive file formats.[16] Therefore, gvSIG 3D seems to be not very useful for this project.

[13] http://resources.arcgis.com/en/help/main/10.2/index.html#//015w0000003z000000
[14] http://www.gvsig.org/web
[15] http://gvsig3d.blogspot.nl/
[16] http://devel.gvsig.org/download/projects/3d/dists/0.2.0/docs/gvSIG-1_10-3D-0_2-man-v1-en.pdf

Google Earth[17]

Google Earth is freeware (so by definition, it is closed-source). "Google Earth is a [3D] virtual globe, map and geographical information program."[18]

- Google Earth contains a 3D viewer with basic 3D functionalities for navigating.
- Google Earth has a nice example of an 'historical layers' tool, where a scroll bar is used to view the earth as it was in earlier periods. Comparable to our '**historical reconstructions scroll bar**'.
- Google Earth has a nice example of a '**time & daylight'** tool.
- Data can be imported into Google Earth via KML files. KML files only support polylines and polygons though, not point cloud data. Therefore it seems *unlikely* that Google Earth can be a base for the user interface.
- Although Google Earth is freeware, Google Earth's browser plugin-specific **interfaces** are available on line: https://developers.google.com/earth/documentation/reference/?hl=nl

Conclusion

The most common 3D GIS software (ESRI ArcScene), for which Mapping the Via Appia has a license, and open-source 3D GIS software (gvSIG) do not have the necessary functionalities to fulfil this project's needs. It seems that those functionalities (point cloud alignment, segmentation, database connection & query, historical pictures (see "User requirements" document)) need to be integrated into existing and extendible 3D software. CloudCompare appears to be a good candidate. It is open-source and extendible, contains all needed basic 3D functionalities, has a 3D measure tool, and even a point cloud alignment tool (not fully automatic though). It also supports the import of archaeological reconstruction meshes. Furthermore, it allows for PointCloudLibrary plugins, where segmentation and alignment libraries can be found. Other sources or inspirations for the functionalities that are to be integrated can be found in MeshLab, 123D Catch, Google Earth, ArcScene and Fugro Viewer.

[17] earth.google.com
[18] http://en.wikipedia.org/wiki/Google_Earth