

Graph-based Guidance in Huge Point Clouds

Claus SCHEIBLAUER / Michael WIMMER

Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria

Abstract: In recent years the use of remote sensing devices like laser scanners in the documentation of excavation sites or cultural heritage sites has led to huge point cloud models from these sites. These data sets may cover complete sites including galleries, corridors, halls, and open places. Orienting oneself in the point cloud becomes a challenge, if one is not familiar with the layout of the site. Therefore we propose a graph-based guidance system to show tourists round the point cloud models. The tourists can navigate interactively through the point cloud, but they are tied to a predefined 3D graph which represents the possible ways, and which connects the points of interest.

Keywords: navigation, information system, out-of-core, point clouds.

Introduction

In recent years laser scanning has become an ordinary tool in the process of documenting excavation sites or cultural heritage sites. Due to advancements in laser scanning technology, point models of these sites can be produced in shorter time with higher accuracy and point density, compared to the previous generation of laser scanners (RIEGL 2013). The resulting point models can consist of several billions (10^9) of points,

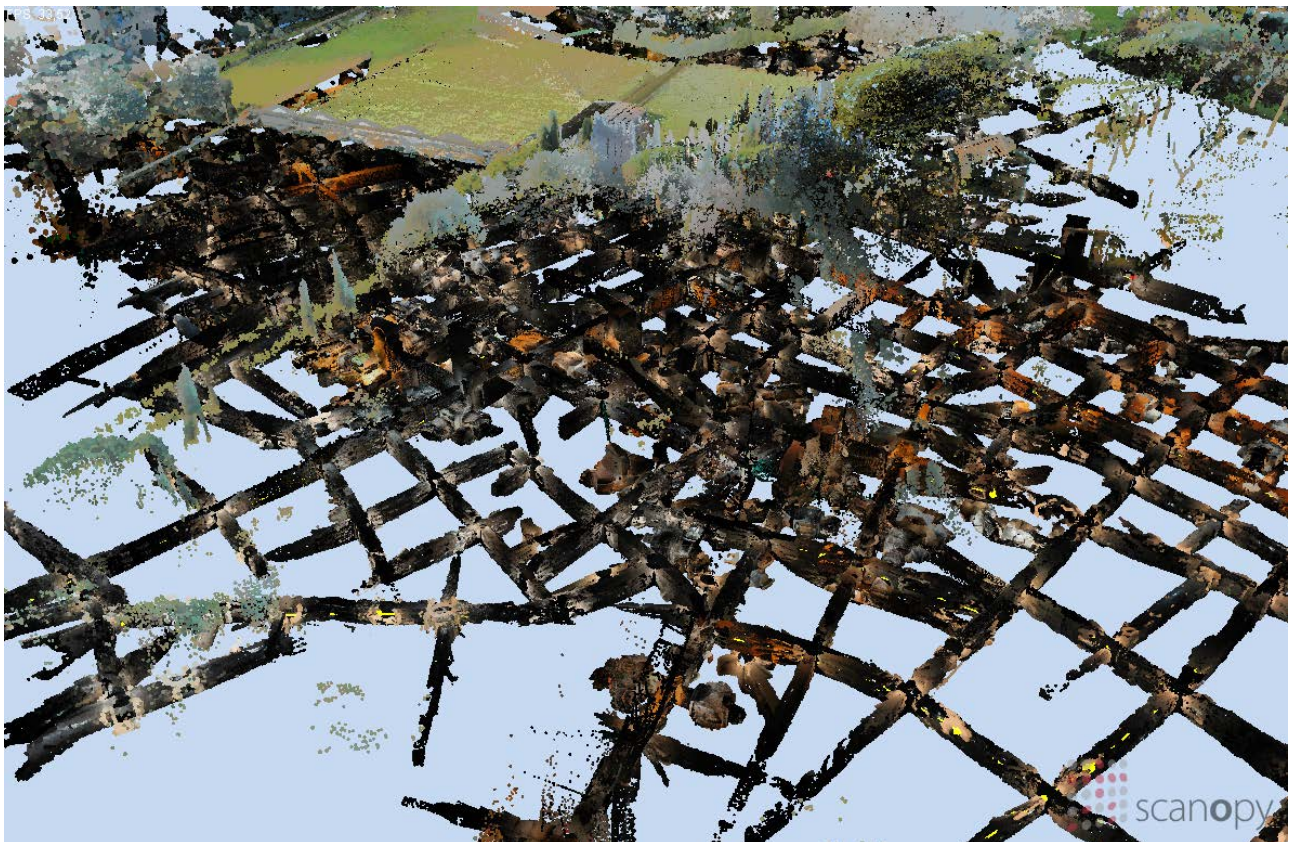


Fig 1 – Point model of the Domitilla catacomb in Rome.

capturing not only complete excavation or cultural heritage sites, but also the surrounding areas. The dimensions of these sites can be quite large, with galleries and hallways up to several kilometers, so navigation and finding a way around these sites can become a problem of its own.

For the purpose of navigating within huge point clouds, we developed a graph-based guidance system. It serves two purposes, first it makes the exploration of a point model less intimidating, which might be the case when exploring a large site, and second it connects the most interesting sights and guides the users to these places. The point clouds which the user explores are not pre-rendered, but rendered in real time. The user can look around freely, but the navigation path is tied to a predefined 3D graph representing the possible ways.

Graph-based Guidance System

The problem we are trying to tackle with a graph-based guidance system can best be described by an example. In Fig. 1 a point model of the Domitilla Catacomb in Rome is shown. The labyrinth like structure of the galleries is easily visible, but these structures are normally covered by the ground level as most of the catacomb is dug into subsoil. Only some parts are also visible from above the surface, like the upper part of the basilica. Around the main entrance to the catacomb also the ground level was scanned and it is visible in the point model (the green surface at the top of Fig. 1). In general the Domitilla catacomb is a late Roman early Christian necropolis consisting of about 12 km of galleries underground, distributed on 4 floors (ZIMMERMANN and ESSER 2007). Due to the huge size of the point model one gets easily lost when trying to find the way through the catacomb. This is further complicated due to the uniformity of the galleries, which often show no decorations. The problem of navigation is true in real life as well as during exploration of the point cloud model. What would be helpful when navigating around this point model is a path the user can cling to, so he can avoid parts of these labyrinth like galleries which are not so interesting, and so instead he is guided to the places of interest. To help the exploration of the point cloud model we propose the use of a 3D graph representation of the point cloud model.

Note that the graph model is not a camera path. A camera path has the movement speed and the viewing direction always given, for each position on the camera path, so following a camera path is not an interactive experience at all. A user is tightly bound to the pre-defined angles of view along the camera path, and he cannot explore the world around him. With the graph based guidance the user can look around freely and also choose, at least to some degree, which way he wants to go. The graph representation acts as kind of a rail system on which the user can move forward and backward, but he cannot leave it.

Another difference, comparing the implementation of a camera path and the graph based guidance, is, that a camera path usually is implemented with a spline interpolation (BARTELS, BEATTY and BARSKY 1987) of the camera's trajectory. Splines have the advantage to enable smooth changes of direction and speed during traversal, but at an increased effort during modeling. When placing nodes spatially too close and at the same time changing direction between nodes too much, ringing can occur, which is an undesired behavior. As for a graph-based guidance the prerequisites for an artifact free spline interpolation of the graph

cannot always be met due to constraints of the underlying geometry, we instead use linear interpolation when traversing the graph. Therefore the camera travels exactly along the edges and nodes of the graph.

Target Audience

Before implementing a guidance system we had to define which target audience we want to support by looking at what kind of user types are actually interacting with huge point cloud models. On the one hand we have the expert user. This is the person who knows all about the scanned object, he is maybe the employer who ordered a scan from a specific site, or he is the person who scanned the site. He interacts often with the data to inspect it, to judge the quality of the scans, and so on. He knows the point model, the layout of the geometry, and he also knows the program controls, how to move around and interact with the data.

On the other hand there is the interested layman, a person who can be described as a visitor in an exhibition. He will interact with the data for the first time, so it is probably more difficult for him to orient himself inside the model, and he also does not know where to search for the places of interest in the model. With the graph based guidance we want to aid the interested layman in exploring huge point clouds by presenting the most interesting ways through the data.

Preprocessing

In a preprocessing step a 3D graph is created manually. A graph consists of 2 sets, a set of nodes and a set of edges. Each node is connected to one or more other nodes by an edge. A node represents a junction or a change of direction, whereas edges represent straight galleries. The graph is not limited to lie in a plane, but can also have nodes at higher or lower levels. This gives enough flexibility to model a graph according to the

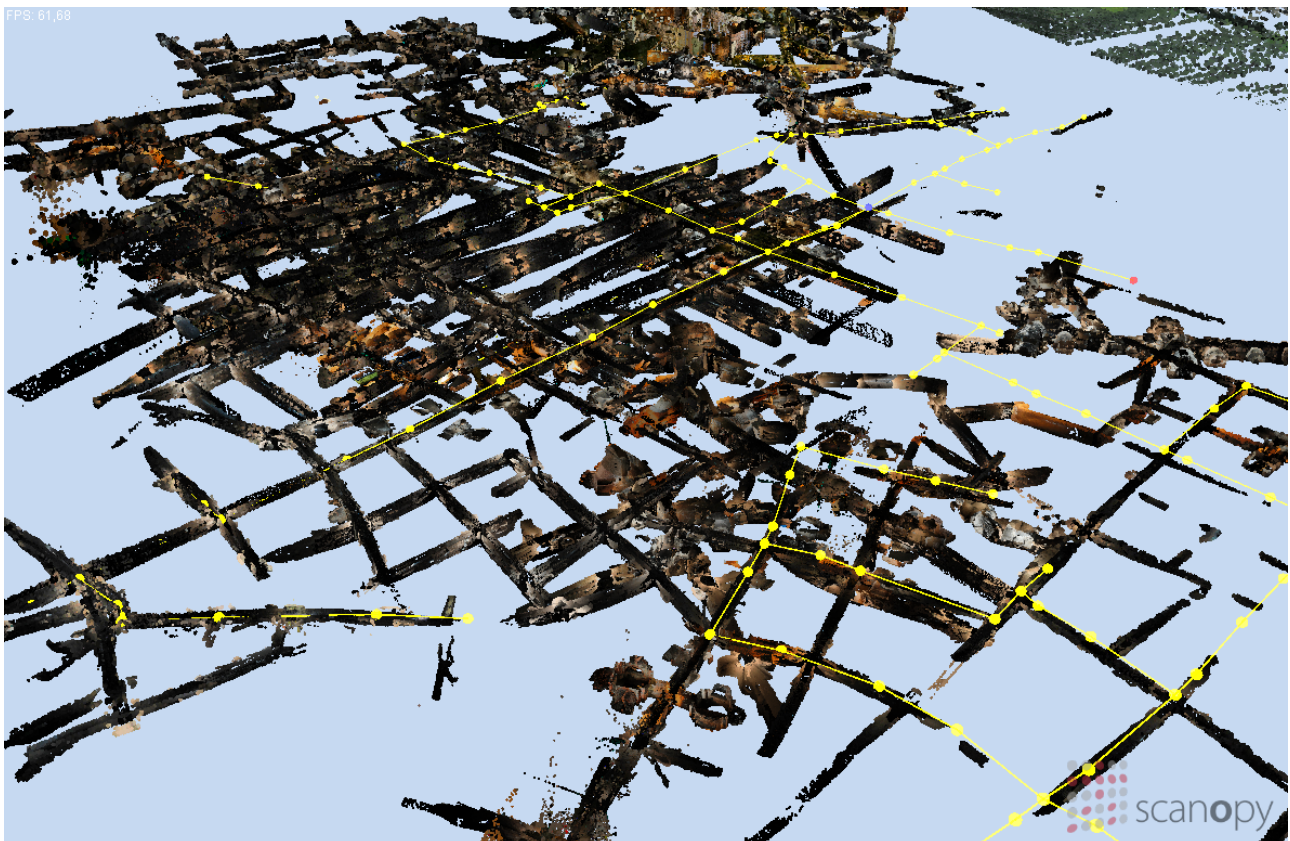


Fig 2 –Cut away view of the Domitilla catacomb point model –with a yellow graph model inside.

layout of the point model. In Fig. 2 a cut-away view from the Domitilla catacomb model is shown. The top of the model is hidden to enable a view into the first floor of galleries. The yellow graph model imitates the galleries of the catacomb. In Fig. 2 not all galleries are part of the graph model, as the graph model only connects the places of interest. It consists currently of about 180 nodes, and covers about 15% of the galleries in the catacomb.

Guidance

The point model of the Domitilla catacomb is especially well suited for some kind of guidance system. Generally speaking, on the one hand, with the usage of the latest generation of laser scanners ever bigger point cloud models are produced. On the other hand interaction with huge point models is still difficult, and orientation inside the point model can become a problem.

We integrate the graph-based guidance system into our point cloud viewer and editor named Scanopy, which enables real-time walkthroughs through point models by employing out-of-core strategies for managing the data (SCHEIBLAUER and WIMMER 2011). By using an out-of-core strategy, all point data is divided up into chunks and usually completely resides on disk. As soon as data is needed for processing, only the necessary chunks of the data are loaded from disk into main memory. When other parts of the data are required for processing, the least recently used chunks are dropped from main memory, thus providing space for new chunks. The same method can be used during rendering a point cloud, as in this case only the data chunks that are visible from the current viewpoint are loaded from disk to memory (and in this case they are then further moved to the graphics card memory). When the viewpoint or the view direction changes, all points that are not visible anymore can be dropped, and the now visible points are loaded from

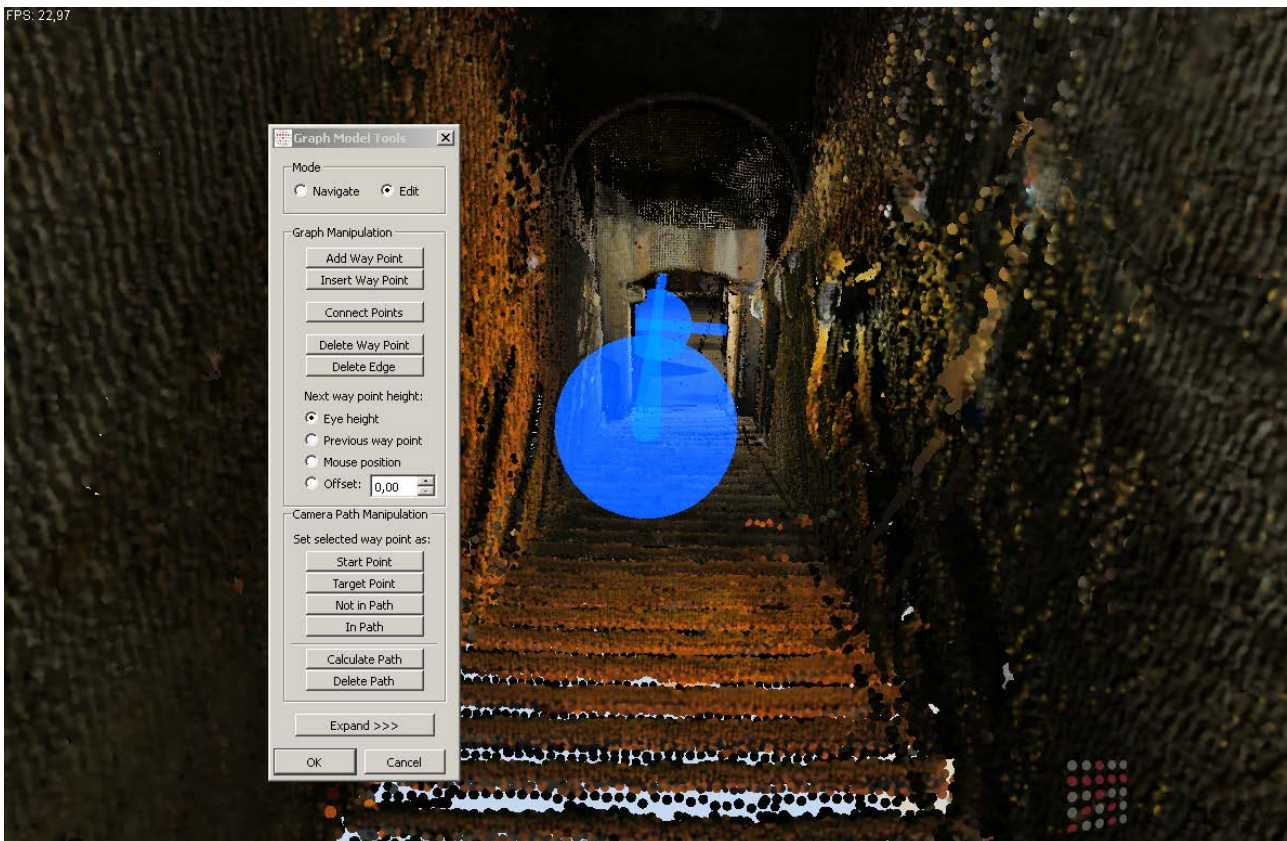


Fig 3 – Editing options for a graph model.

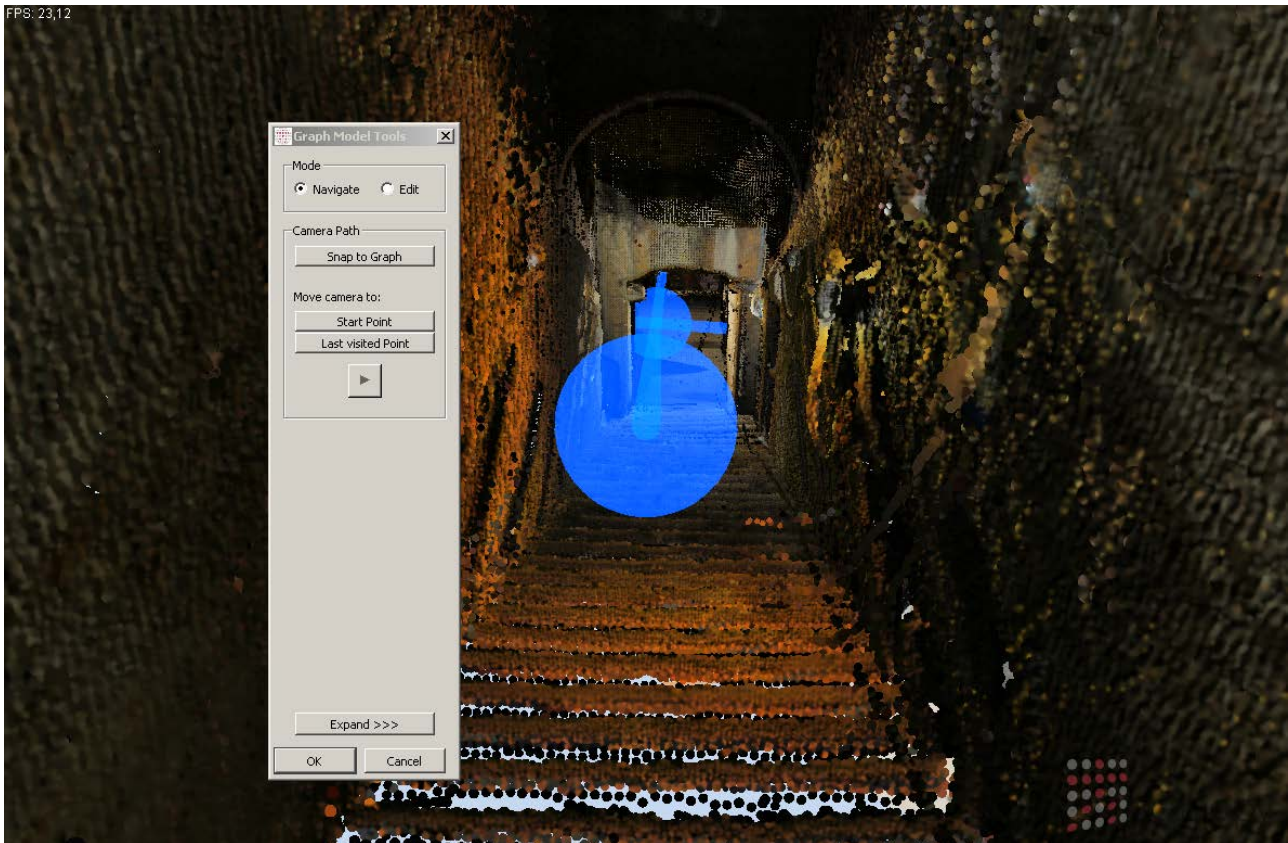


Fig 4 – Navigation options for a graph model.

disk.

In Fig. 3 the user interface for the editing options for a graph model is shown. During editing a graph model the user moves freely around the point model and sets so called “way points”. The way points are the nodes of the graph, and represent either a junction or a change of direction along the graph. Way points can be added (i.e., a new edge connects to the new way point), inserted into an existing edge, or deleted. Furthermore two arbitrary way points can be connected, and edges can be deleted. One problem that occurs during the creation of the graph model, is assigning an appropriate position to the way point in 3D space. This can be done in different ways. During the creation of the graph-based model the user has a sphere shaped cursor that moves along the surface of the point model at the position of the mouse cursor. The problem is that usually the way points are not at the surfaces of the point model but in the center of the galleries in mid-air. Therefore the height of the way point position can be set to be in eye height (i.e., the center of the screen), the height of the previous way point, the position of the mouse pointer, or at an offset to the sphere cursor height.

In the editing options it is also possible to define a camera path on top of the graph model, and this camera path is thought to give an automated movement along the graph model between a user defined start and end point. The path between the start and end point is calculated automatically, and always the shortest path is chosen.



Fig 5 – Choices of direction at a junction when travelling along a graph model.

In Fig. 4 the user interface for moving along a graph model is visible. The button “Snap to Graph” causes the viewpoint to smoothly move to the closest position on the graph model that can be reached. The other two options move the viewpoint to the start of the camera path and to the last visited way point, respectively. When pressing the “Play” button, the camera starts moving along the graph model, while the user can control the speed of the camera movement and the viewing direction by moving the mouse. It is also possible to reverse the movement of the camera by pressing the “UP” and “DOWN” arrow keys on the keyboard.

In Fig. 5 the options when reaching a junction on the graph model are shown. The movement of the camera stops, and the user can choose with the “LEFT” and “RIGHT” arrow keys or with the mouse which path to follow. The view direction is then turning around to the next possible path. Having chosen the desired direction, he can press the “UP” key or click the “Move Forward” arrow on the screen with the mouse to start moving forward. The edge of the path in this particular example is shown in semi-transparent red, but this can be changed to a different color or transparency. The diameter of the graph models edges can also be changed, even until they are of diameter zero, which causes them to disappear. These display options are also available for the nodes, so if nodes and edges are set to zero size, the user will not see the graph model at all.

Fig. 6 shows a point model of the Stephansdom in Vienna, with the viewpoint hovering below the ceiling. The graph model inside the Stephansdom is not only laid out at the height that visitors usually see the interior from, but it also has a path upwards. This is an example of how graph models can also be used to direct tourists along unusual paths, which might reveal never before seen perspectives.

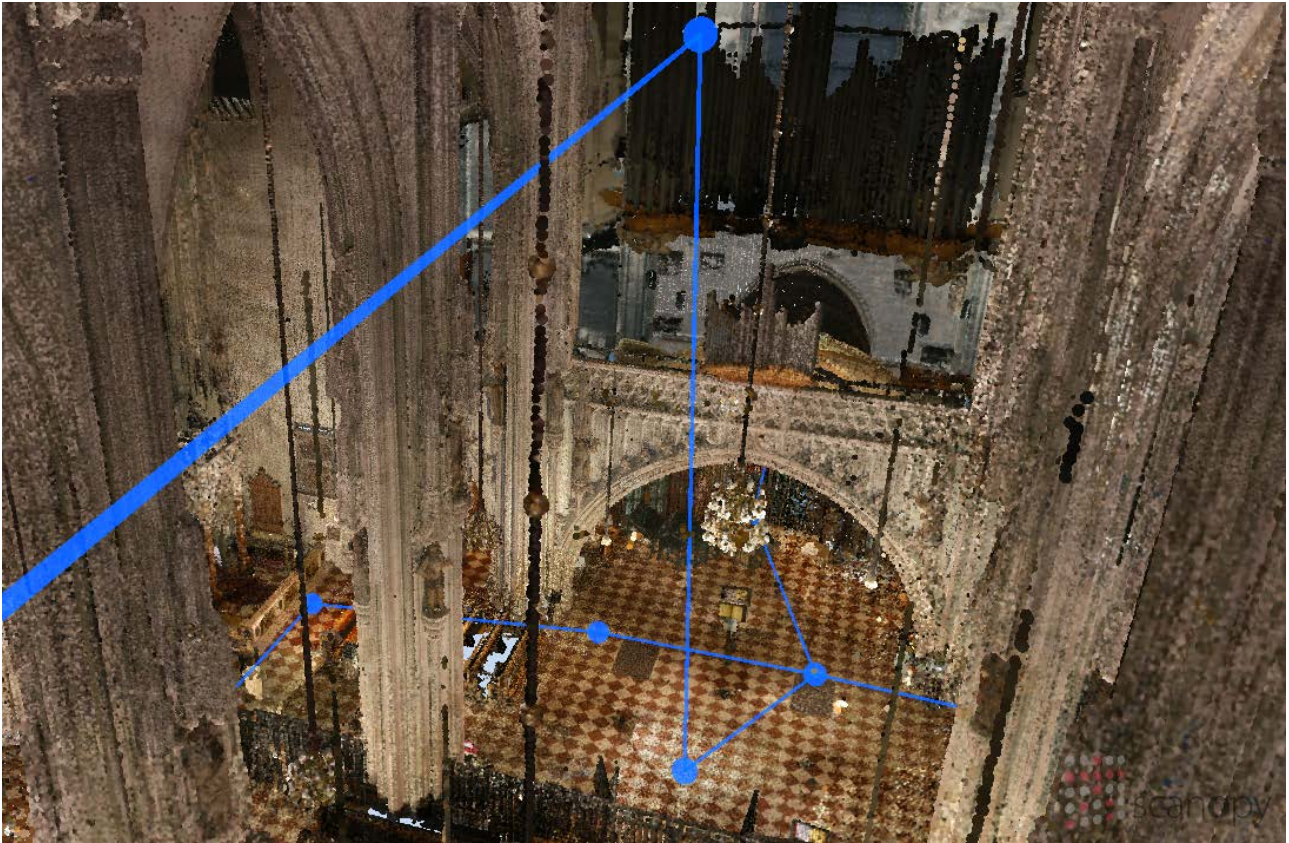


Fig 6 – Interior of the Stephansdom point model with a blue graph model inside.

Conclusion

In this paper we have presented a graph-based guidance system for navigating around huge point-based models. The target audiences for this guidance system are interested laymen or visitors at a museum. The guidance system consists of a 3D graph model of the point cloud, on which the user is bound during navigation, and of an automated navigation system. The user can control the speed at which the camera moves along the graph model and he can look around freely. At junctions he can choose which path to follow for further exploration of the point model.

In the future we want to combine an information system with the graph-based guidance, so that the points of interest are not only visible as part of the point cloud, but they can also be augmented with additional images or descriptions which become visible when the user is approaching them.

Acknowledgements

This work was funded by the Austrian Research Promotion Agency (FFG) through the FIT-IT project “Terapoints”. We thank the FWF funded START project “The Domitilla Catacomb in Rome. Archaeology, Architecture and Art History of a Late Roman Cemetery” for providing the point model of the Domitilla catacomb.

References

BARTELS, R. H., BEATTY, J. C. and BARSKY, B. A. (1987) *An Introduction to the Use of Splines in Computer Graphics and Geometric Modeling*, Los Altos, CA: Morgan Kaufmann.

RIEGL Laser Measurement Systems (2013) <http://riegl.com/>

SCHEIBLAUER, C. and WIMMER, M. (2011) Out-of-Core Selection and Editing of Huge Point Clouds, *Computers & Graphics*, 35(2), 342-351, April.

ZIMMERMANN, N. and ESSER, G. (2007) Showing the invisible - Documentation and Research on the Roman Domitilla Catacomb, based on Image Laser Scanning and 3D Modeling. *Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA)*, 56-64.