3D immersive Underground Navigation

Quoc Dinh Nguyen	Mathieu Bredif	Didier Richard
Univ. Paris-Est, LASTIG	Univ. Paris-Est, LASTIG	Univ. Paris-Est, LASTIG
MATIS, IGN, ENSG	MATIS, IGN, ENSG	MATIS, IGN, ENSG
73 avenue de Paris	73 avenue de Paris	73 avenue de Paris
94160 Saint Mandé, France	94160 Saint Mandé, France	94160 Saint Mandé, France
Quoc-dinh.nguyen@ign.fr	Mathieu.bredif@ign.fr	Didier.richard@ign.fr

Abstract

In this paper, we describe a system for visualizing and exploiting 3D digital data of the Parisian sewer system (PSS). PSS presents a difficult working environment because it is subject to a certain number of risks: toxic gases and waste, biological risks such as viruses and pathogenic bacteria contained in wastewater. Certain areas of PSS are damp, unhealthy, encumbered with waste. In order to improve the working conditions of the sewers and usage of the network in the context of the research for a reduction on exposures to health and safety risks, we developed such system aiming to do the visualization and exploitation of 3D PSS data. Our system allows sewer operators to gather data, information for observation, direct inspection, indirect monitoring and reporting to assess the condition of such sewer system.

Keywords: Immersive navigation, maintaining sewer system, 3D massive model rendering, underground navigation, iTowns+.

1 Introduction

Every day in French communities, millions of gallons of human and industrial wastes are sent though complex underground sewer system the empty into wastewater treatment plants. By example in Paris, the Water and Sanitation Technical Department (STEA) manages a 2400km line of underground sewerage system including 1435 km of sewer; 125 km of collectors, 160 km of manhole which are accessible from the sewer and 514 km of special connections, most of which are closed to sewer.

The PSS are older than 100 years and operates every day, 24h a day, to collects and transports polluted water to treatment plants that clean it and contribute to a healthy environment for our families. Given such age of such sewerage system, the condition assessment of the underground infrastructure is necessary for guarantee and maintaining the proper function of the system in day-to-day operation. In addition, knowing the structural condition of the system will allow us to avoid emergencies, prioritize repairs and replacement projects, and plan for the future visits.

In order to improve the working conditions of the sewers and usage of the network in the context of the research for a reduction in exposures to health and safety risks. We develop a system aiming to do the acquisition, visualisation and the exploitation of 3D PSS data.

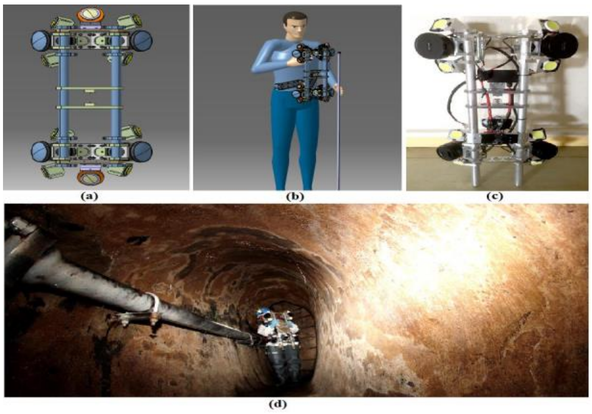
With the scope of this article, we introduce our work by focusing on 3D immersive navigation but the acquisition, which is based on photogrammetry sensors, and the 3D data reconstruction, will also be covered in short in section 2. In section 3, we will explain our 3D immersive navigation. The system allows navigating from the surface to the interior of the sewer, allows realizing a realistic rendering in which any authorized user can move virtually in the 3D geo-referential sewer model, take measurements, make annotations and then to constitute a "prevention" technical file while being easily usable by a person who is not familiar with GIS tools. Section 4 explains our conclusions and future works.

2 Sewer acquisition system

We have witnessed a spectacular growth of 3D reconstruction technologies such as laser scanners (airborne lidars and terrestrial laser scanners) allowing the rapid acquisition of high-resolution 3D topographic data. These techniques spread widely today but investments remains heavy and are suitable for street level data acquisition (Nguyen 2016). For acquisition condition in sewer, photogrammetry sensors are most suited because of their compact size and low power consumption; in addition, making 3D reconstruction tools are very accessible such as Photosynth (Microsoft), Photofly (Autodesk), Bundler-PMVS Arc3D webservice tools or MicMac (Deseilligny 2011).

Our acquisition system composes of number cameras. Among the set of sensors usable for 3D reconstruction, we chose photogrammetry sensors because they seem to us the most suited to the condition in sewers such as a compact size and low power consumption. Moreover, there is no need for an inertial unit, an expensive device that consumes a lot of energy and requires frequent shutdowns. Our acquisition, as be illustrated in figure 1 (a), was based on three sensors: two-photogrammetry sensor for 3D sewer model reconstruction and a pressure sensor to measure the depth of the water.

Figure 1: the figures (a)(c) illustrate our acquisition system, the photogrammetric bench is based on 4 IGN camlights, with 8 led bulbs, a control box based on a Teensy microcontroller and a 15V battery. The figure (b) illustrates our depth pressure sensor; the external pressure is compared to that inside the pipe. The figure (d) illustrates a person carrying our acquisition system inside the PSS.



Our acquisition system is modular as it can be reconfigured with number of cameras, base, etc. The measurements must be as stable and accurate as possible to avoid the error accumulation. For this delicate task, we chose cameras developed in the LOEMI laboratory of the IGN that is pre-designed to drone photogrammetry with many advantages such as global shutter, geometric stability, low mass and low consumption, optical quality and radiometric, etc. This camera is called "camlight". The cameras allow a complete control of their configuration and their embedded software. A radiometric calibration was carried out on each camera before an acquisition to integrate in the hardware processes corrections of linearization of the sensitivity of the sensors.

2.1 3D point cloud reconstruction

We use MicMac tool, which is developed by IGN to calculate a depth map from wide large-scale oriented image dataset. The general pipeline of MicMac 3D modelization process is quite classical and divided in 3 main steps:

1. Compute SIFT descriptors and search for tie points from all pair of images to find POI correspondence between selected image pairs. Unconnected images are eliminated from others by using a filter.
2. Compute relative orientations from these tie points and convert the relative orientation in absolute orientation with the help of aero triangulation calculation on the calibrated images. The data are geo-referenced in RGF93.
3. Compute the dense matching from the oriented images to have point clouds.

2.2 3D mesh triangulation

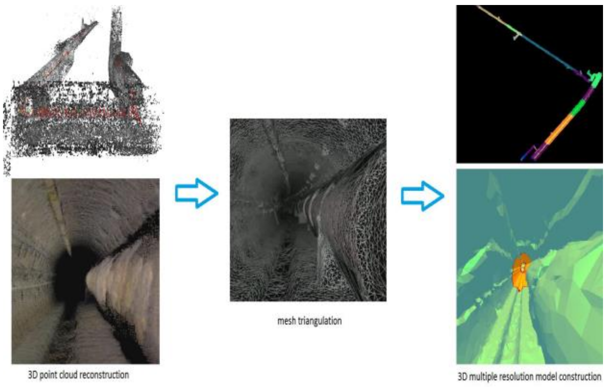
In order to generate a surface from the 3D point clouds, we reuse the method based on the segmentation of the 3D space proposed by (Caraffa 2016). The method consists of segmenting the space into inner and outer zones (the reconstructed surface being the interface between these zones). Instead of choosing a strong surface prior and specific output, the method produces a segmentation of the space related to the confidence to be empty or occupied.

2.3 3D multipe resolution model construction

The 3D sewer models for a city often is hundreds of gigabytes large, it comes as no surprise that the entire dataset cannot be in client memory all at once. At this size, no web-based application can cope with that demands us to create a hierarchical data structure that has motivated the use of out-of-core rendering algorithms.

When we navigate inside a big mesh, some parts of mesh may be occluded or outside the camera view frustum and thus do not contribute to the rendered image. In addition, some regions of mesh may lie closer to the viewer than others; so separated parts of a mesh need to be rendered at different levels of details. This problem can be addressed by creating a mesh hierarchy.

Figure 2: multiple resolution model construction, the point clouds are reconstructed from calibrated image by MicMac, which are used to generate a high resolution mesh, then a multiple resolution model is constructed



We develop our mesh hierarchy based on Kd-tree chunk-based multiple resolutions data structure. The chunk is optimized in a GPU friendly fashion with a triangular-stripe layout and is thus ready to send to GPU memory on demand. The Kd-tree root node contains a drastically simplified version of its original mesh. Each child node contains a subset of its parent, where each subset is more detailed than its parent but covers a smaller spatial extent. The union of the meshes borne by leaves of Kd-tree (maximum depth) encodes the original mesh at full resolution. When rendering, GPU only needs to consider each chunk rather than considering individual triangles, as required in state of the art algorithms.

The data structure reduces complexity of the geometry being rendered and also allows us to develop a view-independent out-of-core viewer to filter out as efficiently as possible the data that is not contributing to the rendered image and cope efficiently with the insufficient amount of GPU and CPU memory. It also enables a progressive transmission of sewer meshes; reduces loading time with data compression; eliminates decode time by decoupling render thread and decode thread and is GPU friendly so the data can be uploaded directly to GPU after decompression.

2.4 Data format

We want to have a format, which is efficient to hierarchy data access; efficient to stream and communicate them to the graphic card at full GPU speed and rendering them in the least necessary number of WebGL draw call. We decide to store the data in a binary format in little endian, with two main parts: a header which contains the metadata that helps the out-of-core algorithm access easily to data and a body that contains the data and can be accessed in JavaScript as an ArrayBuffer.

The structure of our data format is illustrated as in figure 3. The magic number helps us to define data format name, total vertices and faces number let us know number of vertices and faces that are contained in the file, these parameters will be used in out-of-core algorithm to estimate the performance of rendering and to adapt the rendering to different devices. The correct geographic coordinates system requires accuracy and precision in order to match real world locations, usually the vertex coordinates must be stored as 64 bits float numbers to address accuracy and precision needed for such datasets, but it make the dataset a double of storage in relation to 32 bits float numbers. In our format, we store the vertex coordinates safely as 32 bits float numbers, which is completed by a pivot of 64 bits, float numbers in rendering.

The horizontal culling point is not necessary for this work but it will be used in our future work when we render sewer model on globe. When it comes to rendering on a globe, we need to consider the Earth itself as an occlude especially when the viewer is close and parallel to terrain surface, horizon culling really eliminates many occluded tiles.

The node array contains every node information of Kd-tree, for each node, we can access to its child area by using chunk.

Figure 3: Data format for multi-resolution meshes.

header

Magic	Version	Vertex Number	Face Number	Node Number
(uint32)	(uint32)	(uint64)	(uint64)	(uint32)

header

Batch Number	Pivot	HC	BBox	Node Array	Chunk Array
(uint32)	(3xdouble)	(3xdouble)	(4xfloat)		

node array

Attribute	ByteLength
Offset	uint32
Lastbyte	uint32
Vertex Number	uint16
Face Number	uint16
Geometric Error	float
Bbox	4xfloat
First Chunk	
Last Chunk	
HC point	3xdouble

chunk

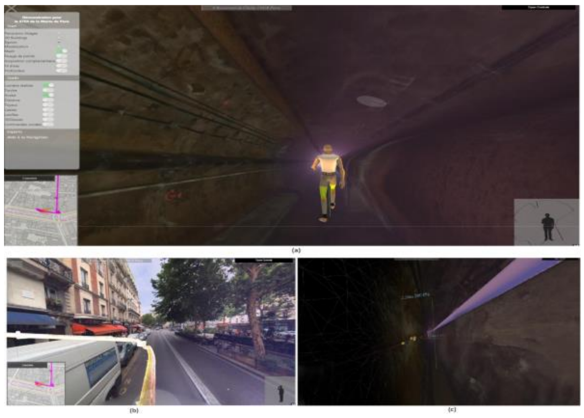
Attribute	Byte Length
Chunk ID	uint32
Last triangle	uint32
Texture ID	uint

3 Immersive underground navigation

Considering that 3D sewer models for a city are often measured in hundreds of gigabytes, it is not possible to render a wide city area with high-resolution models because the amount of data preventing us either to render them directly or to keep them in main memory. The idea is to visualize only mesh portion (chunks), which have compatible viewing parameters (ex. view frustum, SSE). By using viewing parameters, new portions (chunks) are brought into system memory, and old portions are removed, ideally without stuttering rendering. This allows us to keep only small subset chunks in system memory. The rest resides in a file in secondary storage on network server. The strategies are called “Out-of-core”.

We develop the viewing dependent algorithm based on Itowns+. We have chosen this open-source framework because iTowns is the only WebGL-based immersive navigation that is developed under ThreeJS library supported by a large WebGL/Javascript community. Itowns+ has two versions, the first one is developed for street immersive navigation and the second is developed for the globe visualization. In scope of this paper, we chose the first version of iTowns to develop our immersive underground navigation.

Figure 4: immersive navigation inside Parisian sewer system



3.1 Out-of-core algorithm

We extended a view-dependent algorithm, called “chunked LOD”, and originally proposed in Ulrich 2002, for massive terrain rendering. The algorithm operates on Kd-tree sewer mesh to check which chunks are visible from the camera with two conditions, frustum visibility and SSE. Thanks to the pre-computation of viewing parameters, we do not need to load chunk data to estimate these parameters. The viewing parameters are saved as metadata in file header and loaded only one time when chunk is known to be visible from the camera.

We start at root node, according to Screen Space Error (SSE), if the node has sufficient detail for the scene, it is rendered. Otherwise, the node is refined, meaning that its children are considered for rendering. This process goes on recursively until the entire visibility mesh portions are rendered. Unlike original Chunked-LOD algorithm which needs data to be available in client memory to compute viewing parameters, our method runs very fast because the viewing parameters are pre-computed. Only relevant chunks, which are added to the render list, must be downloaded from server. Figure 4 (a) illustrates an example when we navigate inside sewer 3D model with an avatar.

3.2 Projective texturing

Images captured from our acquisition have two usages, we first use them to reconstruct 3D sewer models because the cameras have been calibrated; second, we use these images to texture our mesh on the fly when we navigate in sewer models.

Standard 3D formats (such as OBJ, PLY, 3DS) use UV texture mapping technic, which are stored for each vertex coordinates and mesh responsible a UV coordinates and material to indicate which pixel from the image corresponding to the vertex. Here we do not use UV coordinates to texture the sewer mesh. We load oriented images, which have absolute position information. Since the vertex coordinates are geo-referenced we then know the correspondence between each pixel and vertex in 3D space. The color vertex is also used to increase contrast in zone where oriented image contains fewer pixels. We can then apply a projective texturing technic as follow:

1. Get the chunks around the viewing position by using the out-of-cores algorithm.
2. Get the images and correspondent calibration parameters near to viewing position.
3. Initialize the depth buffer of each image texture with a depth-pass on the mesh and update it as new chunks are received.
4. Blending color vertex with oriented texture image to increase contrast in dark area.

Compared to other traditional texture mapping approaches, we propose three extensions. First, one addresses the web context, as the projective texture calibrations (position, rotation and projection matrices) are stored in a PostgreSQL/PostGIS database with a Node.js front-end, thus enabling fast queries for nearest textures from the current view position and pre-fetching based on query regions. The second one, as we do not texture a simple mesh, but a multiple resolutions mesh, when rendered mesh is view dependent and composes of different chunk LODs. In the third extension, we couple color vertex with oriented texture image to increase dark zones. As illustrated in figure 4 (a), the zone in front of avatar has more eclarage with help of color vertex.

3.3 Click and go navigation

Our system offers the possibility to walk on terrain surface or to go down under into sewers with the help of a click-and-go function (see figure 4 (b)). With just a click, you can navigate to a new place and see this place in panoramas. The street-level and sewer-level imagery are both projected on geo-referenced geometries as explained in previous section. When we move our mouse around in a panorama, a ray-casting method is used for detecting ray intersect with geometry, a disk or rectangle attached to the cursor is shown up to not only makes the panorama feel three-dimensional, but also to show us where we can jump to a new panorama to get a different view.

3.4 Advaned functionalities

To predict the possibility of moving a person in the sewer pipeline we develop an avatar function that allows us to move an avatar within sewer, to identify and to warn obstacles. The avatar can be configured in height to fit an operator whe working under the sewer, it can move along a predefined 3D path or controlled by users. We can also take measurements and annotations to constitue the technical prevention file as illustrated in figure 4 (c).

4 Conclusions

In this paper, we have presented a system, which enables navigating from the street-level to the sewer and visualizing the precise arrangement of sewer and their dimensions provides information obtained during terrain visits. In addition, the 3D geo-referential allows knowing the sewer geometry and exploits the data to be able to realize project studies, 3D views, and to generate cuts, profiles in length that enable visualization and prevent existing hazards before getting real access into the sewer system for the preparation of next visits.

References

Caraffa, L., Brédif, M. & Vallet, B. (2016). "3D Watertight Mesh Generation with Uncertainties from Ubiquitous Data". In: Computer Vision – ACCV 2016. Ed. by Shang-Hong Lai et al. Cham: Springer International Publishing, pp. 377–391. isbn: 978-3-319-54190-7.

Thatcher, U. (2002). "Rendering massive terrains using chunked level of detail control". ACM SIGGRAPH course notes on "Super-size it! Scaling up to Massive Virtual Worlds"

Nguyen, Q., Bredif, M., Richard, D. & Paparoditis, N. (2016). "Progressive Streaming and Massive Rendering of 3D City Models on Web-based Virtual Globe". In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '16. ACM, 83:1–83:4. doi: 10.1145/2996913.2997008.

Deseilligny, M. &Clery I. (2011). "APERO, an open source bundle adjustment software for automatic calibration and orientation of set of images". In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-5/W16, pp 269–276. https://doi.org/10.5194/isprsarchives-XXXVIII-5-W16-269-2011