A 4D visualization tool for TBM worksites using CAP: integration of 3D models and real-time modeling thanks to database connections

Raphaël Gueulet[a], Lionel Milesy[b]

[a]Research and development, Dodin Campenon Bernard, Vinci Construction, France
[b]CAP, Dodin Campenon Bernard, Vinci Construction, France
E-mail: raphael.gueulet@vinci-construction.com, lionel.milesy@vinci-construction.com

Abstract –

The Building Information Modeling (BIM) is getting more and more common into building construction, but it hasn't been developed into underground work so far, although it would be very useful. Indeed, underground works strongly interact with their environment, and with many different stakeholders, whose data are partitioned into their own field. In this paper, a study dedicated to the development of a new decision support system is presented: it is a 3D visualization interface dedicated to worksite's technical management.

This tool is focused on tunnel boring machine (TBM) projects. Its main assets are:

-To have a 4D (3D + time) model, built in real time thanks to the connection to different databases of the worksite.

-To integrate data that comes from different fields such as: TBM excavation parameters, geological data (3D block model describing the lithology and rock alteration), buildings, tunnel as-built, stations, geotechnical and structural monitoring (settlement for example).

Until now, there has been no integration of these data into one single tool, making analyses of their mutual impact on each other quite complicated and tedious.

Keywords –

TBM; 3D modeling; 4D modeling; Visualization; Monitoring; Decision support system; Information management

1 Introduction

CAP is a system for tunnel boring machine (TBM), navigation, guidance, and data acquisition and survey. It is also a subsidiary company whose clients are TBM projects (Vinci and other general contractors).

The navigation module is based on automatic and periodic measurement of the real-time position relative to the tunnel alignment (using a total station).

The guidance module is composed of the steering console controlled by the pilot and the automaton leading to the action of the pushing rams and other actuators.

All the sensors data are stored into a database which can be used for surveying; including real-time analysis and monitoring.

This core system of the TBM is central to the worksite activity. However, it is not the only one needed. Indeed, the geology and geotechnics studies are not included; as well as the structural monitoring (such as the vertical settlement induced by the passage of the TBM).

These other systems are based on their own software solutions, with separate data representation. It means that a TBM worksite is driven by partitioned systems: there is not a global tool to integrate the crucial data throughout the project.

In addition, there is for now a lack in 3D representation for the linear infrastructures, and especially for linear underground works (geometric design tools, and data structures). The creation of Industry Foundation Classes (IFC) dedicated to underground works is at its early stage [1].

These are the reasons why we decided to carry out a study with the aim of developing a new software for CAP system which would integrate data not only related to the TBM, but also to the other systems and available 3D models (for visualization purpose, not for navigation).

The research conducted herein has led to the creation of a 4D (3D + time) interface for data visualization and automatic tunnel modeling. However, it should not be considered as a BIM tool.

It has been developed in Unity3D and is in the prototype and demonstration stage. It has not yet been deployed/marketed on TBM's projects: which is what we aim to do (from one root software, making different projects for each worksite/client).

The data used in this case study is a real project: Rennes (France) metro line B (under excavation between

2015 and February 2018).

The rest of the paper is structured as follow. Section 2 presents the inputs that we wanted to integrate and how we worked on to do so. Section 3 describes the integration processes driven by a maximum of automation. Section 4 describes the implemented tools. Section 5 presents the feedbacks of the operators who would be the users of this system.

2 Inputs

2.1 TBM digital model

For the TBM modeling, 3ds Max has been used for its convenience for computer graphics design. Indeed, the head (cutting wheel, shield, tail, articulation jacks, pushing rams, erector segment) is geometrically conform to the blueprint, but the rest of it is only computer graphics representing an earth pressure TBM. It has then been textured, and rigged to enable its animation.

The Figure 1 represents its head, and shows the cutting wheel, the shield (separated in two by the 14 articulation jacks), and the 14 pushing rams (the jacks pushing on the concrete rings).

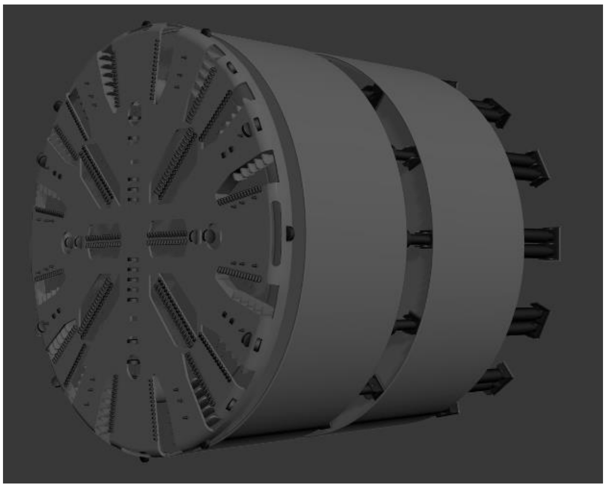

Figure 1 : Model of the TBM head

2.2 TBM parameters

CAP stores all its sensors data (it usually represents around 1000 parameters, including computed ones) in a relational database with a one second timestep. For this study, only around 60 seemed relevant to use.

The data needed for spatial representation of the various parts of the TBM are the articulation jacks and pushing rams extensions; and most important, is the curvilinear abscissa of the cutting wheel on the alignment of the tunnel (chainage).

The other data are the main excavation parameters: energy, speed, torque, speed, jacks pressure, etc.

2.3 Geological block model

The geology of the project has been modelled by interpolation of rocks interface identified into the drilling logs. The limits in between two given layers are considered as 3D points. The process consists in computing an interpolated surface passing through that points: giving a 3D model of the border of that two layers.

This is performed over all the drilling logs description, for each lithological rock interface. Additionally, the rock alteration profile can be modelled.

This leads to the generation of multiple 3D surfaces [2], constrained by the modeler, to its geological and/or geotechnical interpretation: the common constraint is to give a preferential orientation and dip of the layers. This 3D geological output has been generated using Eureka software (Maptek).

For automatic integration of the 3D geological data, it has been chosen not to use these 3D surfaces directly as inputs. Indeed, the objective was to map a mesh (ground) with geological information extracted from the 3D model; the most automatic process is to use a 3D block model composed of voxels (volumetric data). This geological block model has been computed from the 3D surfaces (Eureka software output) using Vulcan software (Maptek). The size of the cells has been set to 0,5m x 10m x 0,5m to fit the orientation and dip of the layers (E-W 70° S).

The Figure 2 shows the alteration profile of: a) two 3D surface meshes interpolated from the drilling logs information b) a cut in the block model obtained in Vulcan.

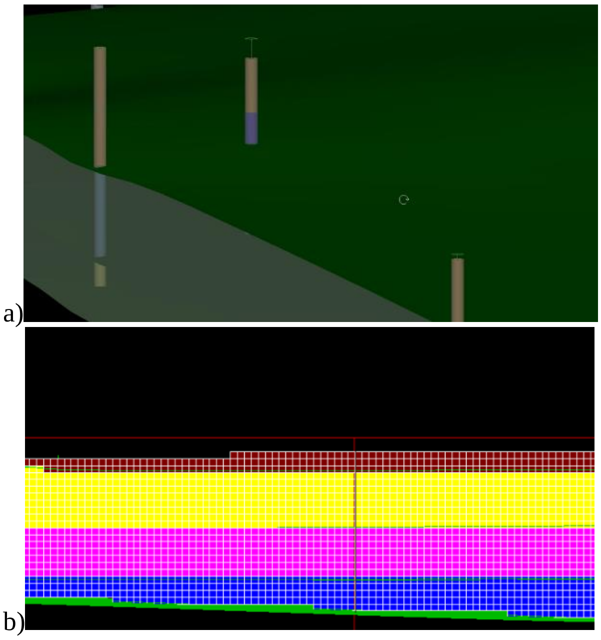

Figure 2 : a) 3D surfaces and drilling log in Eureka b) cut in the 3D block model in Vulcan

2.4 Structural monitoring

The structural and deformation monitoring is crucial for ensuring the stability on top, at the passage of the TBM. It is mainly automated, thanks to the setup of many different sensors (extensometers, inclinometers, total station / prism target etc.). On the ground, vertical settlement and differential vertical settlement is monitored.

Specific companies ensure this monitoring, centralizing in real-time into a database all the sensors data; analyzing them and computing rules and threshold for trigger alerts. They also provide software solutions for communicating to the worksite these monitoring results with GIS maps showing sensors and their values. Arising from this tool, the operators are facing many different maps and graphs, corresponding to different infrastructures, orientation, levels, etc.

We chose to represent only the vertical settlement; in the area where the settlement induced by the passage of the TBM can occur: 30m before the cutting wheel, as far as 100m after the cutting wheel (after the passage).

2.5 Tunnel alignment

A tunnel alignment does not have a 3D geometric description, because it is defined in projection plans. In the horizontal plan (x, y), it is composed of straight segments, clothoid, and circle arc. In the vertical plan, along the tunnel alignment – that is the 2D alignment previously defined – it is composed of straight segments and parabola. This geometric data is what is used to compute the position of the TBM relative to the tunnel alignment (thanks to the topographic polygonation and automatic total station tracking of the TBM).

The first extension for the IFC5 (dedicated to infrastructures) – the IFC Alignment [3] – has not been so recently released, but, to our knowledge, has not been deployed on linear infrastructures projects so far. It is connecting to GIS-like modeling, and will be the base for IFC bridge, IFC rail, IFC road and IFC tunnel [4]. While IFC Bridge is on progress for many years [5], the work on the IFC Tunnel has just started (in 2017 in France, via MINnD national project [1]).

We did not used IFC Alignment format; but simply the geometrical information of the tunnel alignment to compute a series of 3D points. A one-meter resolution of the chainage has been chosen; which is a greatly satisfactory precision: the 300-meter minimum radius resulting in 0,4mm maximum distance between circle arc and one meter segments.

The Figure 3 displays the tunnel alignment made of 3D points, in horizontal plan.


Figure 3 : tunnel alignment in horizontal plan

2.6 Stations BIM models

We had to our disposal the BIM models of the stations in IFC 2x3 format. These are LOD 300 BIM models (level of detail), and are segmented in subtrades which enables the integration of civil works only. These models have been exported via Revit in FBX format, keeping only the geometry and not the metadata. The Figure 4 displays one digital model of the project.

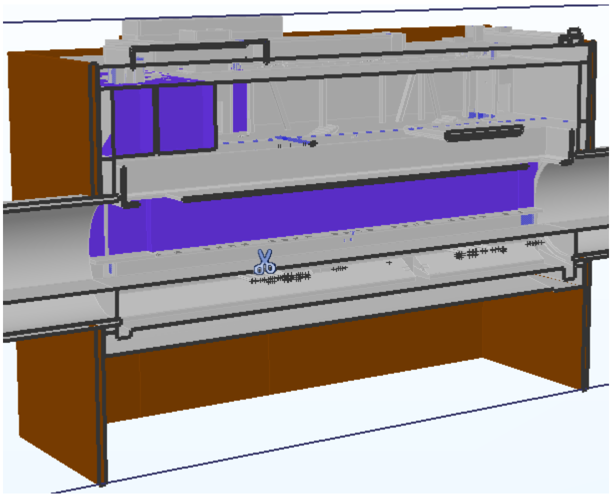
Figure 4 : BIM model of one station

2.7 City buildings model

Virtual 3D city models can be obtained from multiple techniques: photogrammetry (satellite, aerial or close range) and laser scanning (aerial or terrestrial laser scanning) [6].

After acquisition, the modeling can end up on different data: raw meshes, segmented post-processed models (like as-built buildings models with a certain level of detail). The standard file format for geographic information system (GIS) applied to cities is CityGML.

CityGML open-source data, including 3D models, are available for many big cities (for example Lyon metropolis, France [7]). These data are very relevant to integrate in a 3D application since they are segmented in sub elements, which can be used for interaction.

However, for the city of Rennes this data is not available. We decided to use satellite photogrammetry data composed of a unique mesh, including some artifacts, and vegetation (which is not the desired focus).

The Figure 5 shows a part of that mesh.

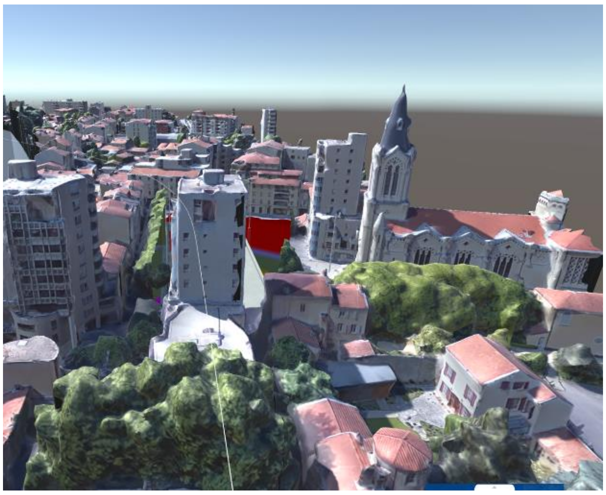

Figure 5 : City mesh

For further development, we would consider using CityGML models for the reasons mentioned above (which, if not provided in open source by cities, can be bought from dedicated companies).

2.8 Ring (tunnel lining segment) digital model

For tunnel modeling, the elementary object is the concrete ring. It has been designed on Inventor, based on the blueprints of the project. Its geometric features are detailed in section 3.3 p.4.

3 Integration into a Unity3D software

3.1 Integration overview

The inputs are integrated into a Unity3D software with maximum automation (for further projects to be created most efficiently). The TBM model is positioned and animated thanks to the TBM parameters, accessed via a http query on CAP database.

The digital models (static) were integrated into Unity3D using FBX format.

The geological block model is at the frontier between static and dynamic data: an update in the model is automatically integrated into the software (part of the Streaming Assets, with no need to get back to the Unity Editor). The methodology for its integration is developed in Section 3.4.

The Figure 6 illustrates the overall integration in the software. The verification is quite basic: the problems we encountered were due to bad georeferenced data (the worksites often use multiple coordinate systems); which are easily detected (visual check).

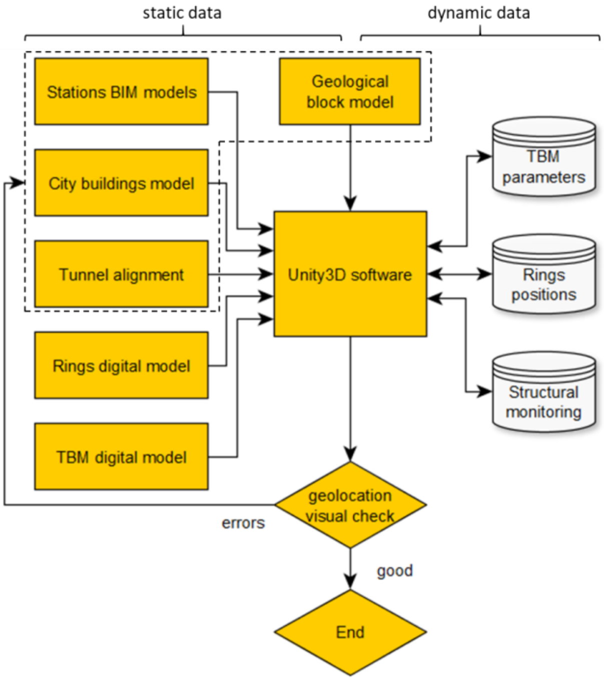

Figure 6: Inputs integration and verification overview

3.2 The choice of Unity3D

Unity3D is a game engine and an integrated development environment used to develop video games and simulations for computers and other devices. It is increasingly used in many industries applications.

Developing plug-ins onto BIM software would have been more complicated than using Unity3D (if not impossible); and it seems more likely to become obsolete with development on a specific software.

There are also convenience reasons that naturally led to choose Unity3D: drag-and-drop functionality, C# scripting. Unity3D being multi-platform is something valuable too: our software is only built on a computer application, but AR or VR may have a use for this tool.

Additionally, its widespread use in the field of 3D industrial application makes it an appropriate solution to plug in with other similar tools.

3.3 Automatic tunnel modeling

The real-time storage of rings position enables the automation of tunnel modeling. The json answer to the http query on the database is structured as follows: ring number, ring type (there are two types of rings: 1,3m thick and 2m thick), and angular position of the key segment. Besides, the chainage of the first ring is given.

A ring is composed of different segments (seven in our studied project). They form a cylinder cut at its borders by planes which are not orthogonal to the axis of the cylinder (forming a trapezium if looked at in the right projected plan, cf. Figure 7).

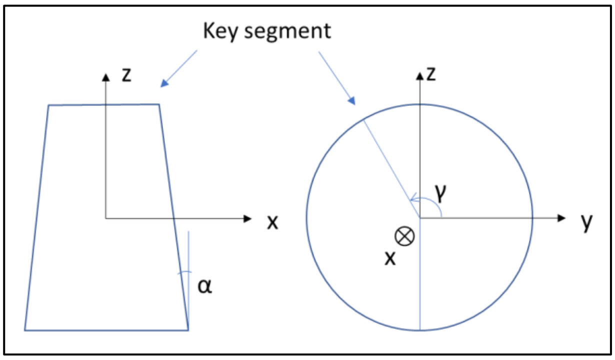

Figure 7: Drawing of a concrete ring

From the chainage of the ring, we can compute five degrees of freedom (DoF) thanks to the tunnel alignment data. The sixth one is given by the angular key segment position (γ) which is incremental.

This information is satisfactory to instantiate the sequence of ring models centred on the as-designed tunnel alignment, and with the true key segment positioning. Since the segment positioning is not defined by the construction survey, we can consider that as being a semi as-built modeling of the tunnel. The scripting of this part, and the frequent database connection (0,1Hz), makes it an automatic and real-time modeling tool.

Moreover, as time information for the pose of rings is stored in the database, we gave the possibility to watch its evolution over time (making it a 4D model). In addition, a metadata has been added on the segment models to link it to the document management system (DMS) of the project relative to the quality survey of the rings and to the traceability of their manufacturing.

The Figure 8 represents a part of a ring sequence seen from inside, and in which we can navigate. The ring number is displayed, as well as the segment numbers (the key segment is displayed with darker color).

Figure 8: Semi-as-built tunnel model

To obtain a full geometrical as-built of the tunnel, the real (as-built) tunnel alignment should be integrated.

M. Lu et al. computed an as-built of the tunnel from the position and orientation of the TBM [8]. However, the tunnel is moving a bit after the TBM passage (it is usually going up at a centimeter order of magnitude, due to the subtraction of the TBM weight). That's why topographic surveys are necessary to precisely measure the as-built tunnel alignment (the deformation of the rings is also measured). These data were unavailable at that time, therefore we tested another technique to obtain the as-built tunnel alignment. It has been computed using only the first ring position (for which the 6 DoFs are known), and the sequence of angular key segment position.

Based on the drawing of Figure 7, we can write the angular differences between ring N and ring N+1 (spherical coordinates) as follow in Equation 1-3.

$$\theta_{N+1} - \theta_N = \alpha(\cos\gamma_N + \cos\gamma_{N+1}) \quad (1)$$

$$\varphi_{N+1} - \varphi_N = -\alpha(\sin\gamma_N + \sin\gamma_{N+1}) \quad (2)$$

$$\alpha = arctan(l/2D) \quad (3)$$

γ being the angular key position; θ being the angle from x in the (x,y) plan of the ring frame; φ being the angle from z; l being the tapering of the ring; and D being the diameter of the ring.

We computed it on a dataset – in the cartesian coordinates system of the tunnel alignment – and compared it with the as-designed tunnel alignment. The Figure 9 shows the difference between these two alignments, along with their distance to the first ring.

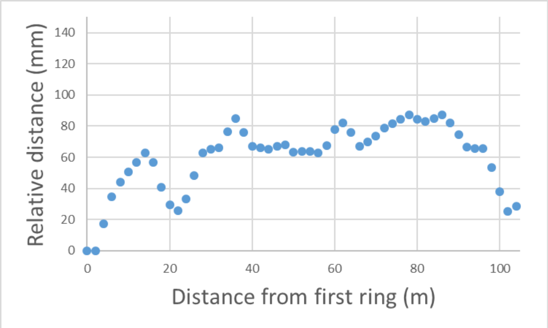

Figure 9: difference between tunnel alignment and computation from ring sequence

These values (around 7cm) are too big to be the true distance between as-built versus as-designed alignment. We suppose that the difference is due to the elasticity at the interface of the rings (rubber sealing).

For as-built alignment computation, we should either use the technique developed by M. Lu et al [8], or wait for the topographic survey to be done. The presented technique is of no use on its own, but could be used in complement with a topographic survey (for computation between the measured rings).

3.4 Creation of a ground mesh with display of geological information

For representing the geological block model data, we choose to create a ground mesh. This method is lighter for rendering computation than a volumetric rendering of the topological data.

The process of the mesh generation is based on “naïve surface nets”, a simplified version of “marching cubes” process [9]. It consists in meshing an implicit surface in a discrete distance field defined in each point of a 3D uniform grid. The size of the grid has been set to the smallest size of the block model cell.

The geometry of the model to mesh is defined by Constructive Solid Geometry (CSG) operator. This simple and fast process consists in combining closed surfaces thanks to Boolean operators (union, intersection, subtraction) [10].

For creating the ground mesh, we combine:

- A cube representing the edges of the ground.
- The surface of the tunnel defined by its polyline dilated to the radius.
- The simplified envelops of the stations by convex decomposition.
- The level at the surface of the ground. The implicit surface of the ground is defined by the distance of a point to the weighted neighbors projected on the weighted normal of the neighbors [11]. The neighbors search is optimized thanks to a KDTree (knnflann).

The surface mesh is then generated by surface nets method; which guaranty one quad per cell, rendering a color information for each cell without the need for creating textures, and wrapping/unwrapping them on the mesh. The drawback is the number of triangle to display, but it is still lighter than a volumetric rendering.

The color information (corresponding to a geological information) is rendered by “vertex color”. We define a color on each vertex of the mesh, thanks to the geological block model, and the color of the triangle is interpolated.

The Figure 10 displays the result of that process on the project dataset, including TBM and tunnel model.

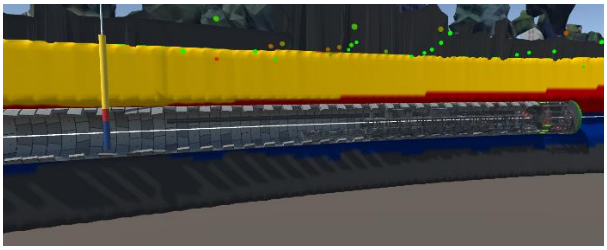

Figure 10: Ground mesh with geological colour information

3.5 Other integrated inputs

The Figure 11 shows the vertical settlement: each monitored target being represented as a sphere colored by its settlement value: green (0mm) to red (-3mm). The big white sphere represents the position of the TBM’s cutting wheel. This is a straight forward integration, relying on a database query.

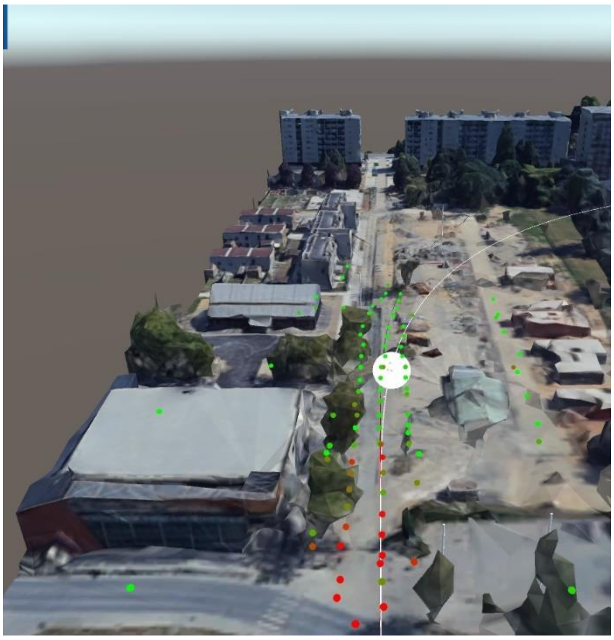

Figure 11: Vertical settlement monitoring

The Figure 12 displays the positioning of the TBM (head and rear cars) thanks to the chainage information, articulation jack extensions, and tunnel alignment. The rings are also represented.

Figure 12: TBM positioning on the tunnel alignment

All this information being stored in databases, it is possible to visualize it over time such as for the tunnel modeling. However, in this prototype software, it has not been fully integrated with data over time. It will need some optimization not to overload the internet bandwidth of worksites with high frequency queries.

3.6 Resource-efficiency compared to data weight.

We worked on optimizing the size of the meshes. The original versions were much heavier than the final ones presented in Table 1.

Table 1. Mesh vertices number classification

Element	Vertices	Triangles
City	1 166 626	2 060 814
Ground	286 167	558 185
Station 1	12 646	51 104
Station 2	78 116	245 079
Station 3	49 293	105 205
Station 4	34 377	94 639
Station 5	27 397	96 300

When testing the run of the built application, it has been noticed that we are far from overloading the processor (Intel Core i7-7500U). Around 40% of its resources are used (cf. Figure 13).

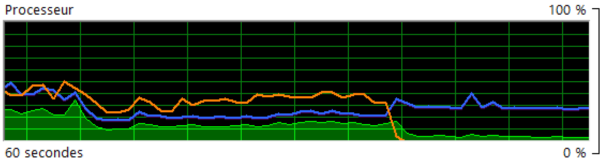

Figure 13: Processor resource (orange curve)

However, a full worksite can be much bigger than this scene (this represents only 1,4km of the 9km of Rennes project). Without optimization on data visualization (such as GIS rendering optimized with the distance of the camera), it will be overloaded. This subject will be addressed on further studies.

These performances do not guarantee every possible project with the same mesh volume: it relies also a lot on other parameters, like the textures. Additionally, a changeover on WebGL could be asked (which would need to do some compromises).

The geological block model represents 87 728 935 points. Instantiating a cube GameObject for each point would have led to over than 2 billion vertices.

4 Implemented tools

4.1 An interpolation of the surface vertical settlement

To give a better representation of the vertical settlement local data, we decided to compute 2D map over the surface of the project, inspired by the interpolations usually employed for level interpolation [12]. The result presented on Figure 14 has been done using an inverse distance weighting interpolation (IWD). We should consider studying also other interpolations to analyze what fits best, because we did not find articles on that specific topic.

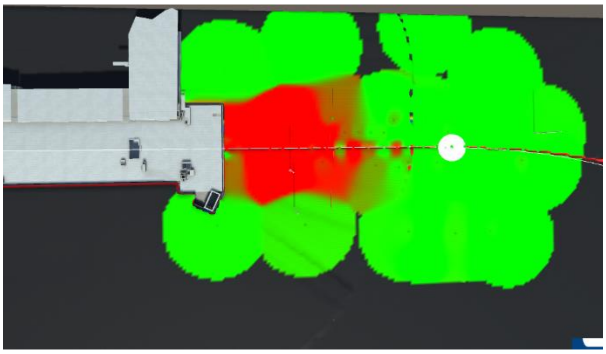

Figure 14: Interpolated vertical settlement map

4.2 A cutting functionality crosswise of the tunnel alignment

Usually, the geological (and potentially geotechnical) drilling logs interpretation does not end up with the computation of a 3D model for underground worksites. In most cases, it is manually interpolated over the longitudinal vertical cut of the tunnel alignment. That’s why the only visualization of the geological data along the longitudinal cut does not have a strong value added.

The crosswise cuts, on the contrary, are not usually produced, except for specific areas where it is required. To differentiate from what already exists, we implemented a cut functionality crosswise of the tunnel, positioned by the user. It is a 2D raster geolocated that gets the data from the 3D block model.

Additionally, it represents the interpolated vertical settlement along its cut segment (the settlement value has been multiplied by 1000 so that it is visible). The Figure 15 displays an example of a cut.

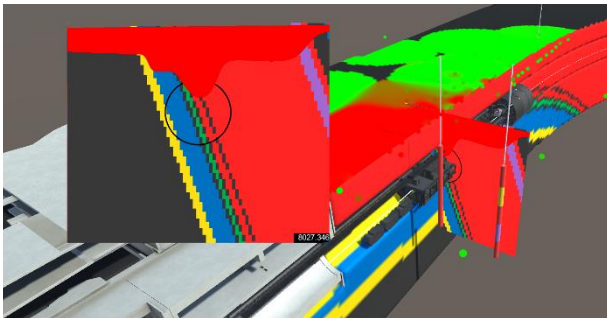

Figure 15: Crosswise cut

5 Feedbacks from the worksite operators

This prototype software has been presented to many worksite operators (technical management) to have their feedbacks. From their point of view, having global information represented in 3D is very valuable. They consider this potential tool as a decision support system that would ease representation of a big amount of data that can't be naturally seen in the real world (because underground). Indeed, the environment, the geology, and the TBM behavior (amongst others) are impacting for the decision making (such as the confinement pressure). Furthermore, the overall visualization can help understand/interpret soil behaviors (e. g. structural monitoring behavior explained by a geological analyze), leading to a better response. It would be particularly helpful when they are facing an incident, and needing to take a quick and efficient decision.

Opinions are diverging when it comes to discuss who the user would be. Everyone agreed that it would be used by technical management, in the office; but some say that it could also be used in the TBM, by the pilot.

Something specific has been noticed, and is important to highlight. From what we know, 3D geological block model is not used in the construction industry (unlike in the oil and gas or mining industries). Operational teams are not used to its scientific representation, and can find it inaccurate since it leads to knurled edges on the cuts, whereas they usually analyze cuts with smoothed interpolation in between drillings logs data. That's why it is necessary to well explain this representation before deploying such data.

The interviewed people also agreed on its limitations: it will not substitute from the other data representations, and software solutions. This new tool aims to ease the overall comprehension of the worksite and better its representation for the operators. It will not enable high end analysis of specific data, for which the existing tools are satisfactory.

6 Conclusion

This study resulted in the development of a functional prototype software of a 4D visualization tool dedicated to underground projects including a TBM: merging data that are usually partitioned into their own field.

We consider that it is a successful proof of concept of what can be done using the data to our disposal from worksites (3D models, and database information) for global 3D integration thanks to Unity3D. The main limitation to broadly deploy this software on worksites equipped with CAP system is the need for manual integration of the data; despite the efforts for maximum automation (such as station texturing, and database connections). This will be further investigated for creation of new projects quickly and efficiently.

One of the problem we faced during that study is the lack of clean, structured data; in addition to the multiple coordinate systems used. For further work – development or deployment – we would ensure that the exchanged data with the worksite are well specified, until BIM become widespread in the underground projects.

References

[1] MINnD IFC extension to the underground infrastructures, On-line: http://www.mindd.fr/activites/extension-ifc-aux-infrastructures-souterraines-ifc-ist/, Accessed 17/01/2018.

[2] Caballero M. Maptek Vulcan Introduction to Vulcan Version 8.1, pages 189-203, 2012.

[3] BuildingSMART IFC Alignment extensions, On-line: http://www.buildingsmart-tech.org/mvd/review/extension/alignment/candidate/html/link/infrastructure.htm, Accessed 12/11/2017.

[4] Liebich T., IFC Alignment Project, Process Map and Use Cases, *buildingSMART*, 2014.

[5] Benning P. and Castaing C. Interoperable information modeling for sustainable infrastuctures – Summary of phase 1. *MINnD National Project*, 10/06/2016.

[6] Singh S. P., Jain K., and Mandla V. R., Virtual 3D city modeling: techniques and applications In *ISPRS 8th 3DGeoInfo Conference*, Istanbul, Turkey, 2013.

[7] Grand Lyon, On-line: https://data.grandlyon.com/, Accessed 10/01/2018.

[8] Lu M., Wu X., Mao S., and Shen X., Real-time as-built tunnel product modelling and visualization by tracking tunnel boring machines In *ISARC*, Montréal, Canada, 2013.

[9] Unknown, Smooth Voxel Terrain (Part 2), On-line: https://0fps.net/2012/07/12/smooth-voxel-terrain-part-2/, Accessed 19/01/2018.

[10] Frisken S. F. and Perry R. N., Designing with Distance Fields, *MITSUBISHI ELECTRIC RESEARCH LABORATORIES*, 2006.

[11] Alexa M. and Adamson A., On Normals and Projection Operators for Surfaces Defined by Point Sets In *Eurographics Symposium on Point-Based Graphics*, 2004.

[12] El Halgawy M., Interpolating Surfaces in ArcGIS Spatial Analyst, *ArcUser*, 2004.