

Roadwork Site 3D Virtual Visualization Using Open Source Game Engine and Open Information Transfer

T. Makkonen^a, R. Heikkilä^a, A. Kaaranka^a, K. Nevala^a

^aConstruction Technology Research Center, University of Oulu, Finland

E-mail: tomi.makkonen@oulu.fi, rauno.heikkila@oulu.fi, annemari.kaaranka@oulu.fi, kalervo.nevala@oulu.fi

Abstract

Machine control systems generate huge amounts of data to be used to build smarter. This information is still not widely used as there are no general methods to access this massive spread of data as these are strictly closed systems. Main tools in our study were an open source 3D game and simulation engine for visual 3D presentation (Panda3D), a machine control system on an excavator which was opened by manufacturer for data transfer (Novatron) and a communication link between these main components Extensible Messaging and Presence Protocol (XMPP). Visualization was performed in 3D presenting model of the road layers and the excavator working on the site using GPS positioning transferred in XML through XMPP. The result shows how to use open source tools to successfully manage data transfer and data presentation using easily accessible open source software and also show potential what these tools could provide for construction software development. For future we hope that every machine control system used will send the data they gather by open methods. This openness should be even demanded as it would most likely generate new ways to use the data and thus generate new business opportunities and would produce new tools to gain higher productivity.

Keywords

open data transfer; 3D visualizations; 4D visualization; machine control system

1 Introduction

Machine control systems for construction machines, like those used in excavators, can be seen as a passive information presenters for a driver, but fast progress on implementing new tools are in a way. Topcon has a very advanced product sitelink3D [1] which, for example, uses machine control system data to show graphically the difference between planned and

processed surface. Novatron's system [2] can send the joint data through internet and work file is sent automatically to the machines.

There is a catch however. These systems are usually closed in nature, for example Topcon's systems are planned to work only with their products. For this approach to work every machine on the site should use only one brand of machine control system, including subcontractors. Infrakit [3] demonstrates how small company can rapidly bring new innovations to practical use on construction site when communications, mainly machine control systems, are opened. Another good example is SVS innovations' Field3D [4] which rapidly brought visualization of building construction project to tablets, a task only possible because of open IFC standard [5].

The work here studies how hard and what is needed to set up communication channel through internet and link between machine control system and open source 3D game engine, to enable real-time visualization of an excavator working on the site. Real-time 3D construction machine visualization is widely studied [6],[7], [8].

For added depth, we study the possibilities of building new kind of tools easily for construction industry, a rude 4D visualization of the work schedule of a road site pre construction is performed. For other 4D related construction visualizations following resources are presented [9], [10], [11], [12].

2 Roadwork visualization

2.1 Experiment setup

Main components used in this research:

1. Extensible Messaging and Presence Protocol (XMPP), originally called Jabber, for information exchange, open standard

2. Machine control system Novatron Vision3D for an excavator, real time communications opened using XMPP.
3. Panda3D game engine, programmed with python language, open source
4. SLeekXMPP a python XMPP library, MIT license
5. Eclipse python programming environment, open source
6. Blender 3D computer graphics software, open source
7. Matlab, proprietary software
8. Tekla Civil, proprietary software

2.1.1 Building communications

The machine control system was reprogrammed by the company Novatron to send data using Extensible Messaging and Presence Protocol (XMPP) [13]. Protocol is build using XML and the form is decentralized client – server architecture. For the work here the most important transferred information is GPS position data, but also data about the construction project like vehicles on the project and organization are included.

On Eclipse IDE SDK 3.7 with python, communication software was programmed using SleekXMPP library [14]. The main components of the program are:

1. Handshaking
2. Registering to a project with a password
3. Reading GPS position data from the wanted construction machine
4. Relaying the GPS position data for virtualization

2.1.2 Building visualization

A 3D-road with four layers and a second one with one layer was constructed using Tekla Civil. The lengths of the road were 50 m and 30 m and the width about 5 m. For visualization the layers were divided to shorter pieces as it would give more possibilities to present visual information. For example area under work can be highlighted in more detailed manner than just the whole layer.

The dividing the road layers to shorter pieces is done semi automatically. From Tekla Civil DXF-file was exported to AutoCAD where road central line was divided to short intervals. The more logical choice of the file format open standard Inframodel [15] was omitted because in theory all the tools to build models directly with DXF were in use without extra programming, however complications arise and open standard should have been used. DXF-file with layers and the central line points where then imported to Matlab where algorithm was created to divide layers to triangle groups. In Figure 1 we can see a road layer

regrouping: Red rhombus marks a centre points of triangles forming a layer, green lines perpendicular to centreline ending to red x marks the selection box, centreline is formed using points acquired from AutoCAD and are used to determine the length of the layer triangle selection box.

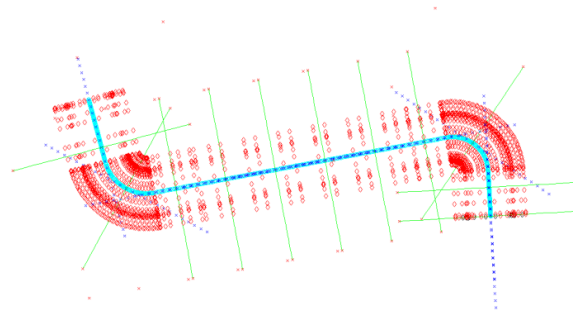


Figure 1. Dividing road layer to shorter parts

To present road layers in game engine Panda3D [16] accurately build-in data format `GeomVertexData` was used. Using the import tools through Blender [17] to Panda3D resulted not eye candy enough results, as seen in Figure 2, where top we see a tool chain import to Panda3D and bottom use of build-in data format `GeomVertexData`.

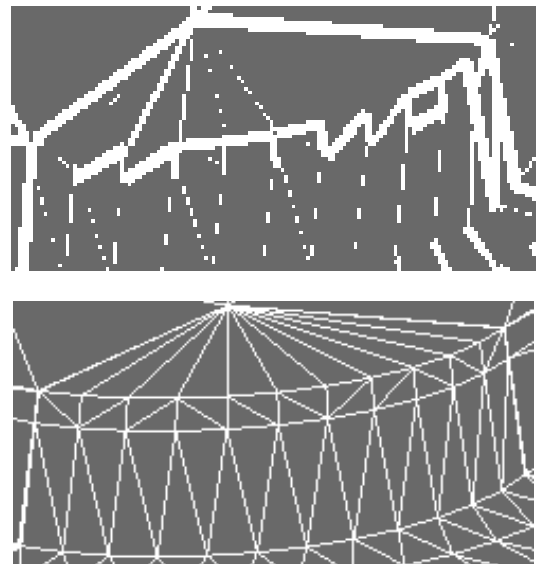


Figure 2. Comparing import methods

At the time of work 3D objects were taken from the Google SketchUp's 3D Warehouse which is today called Triple SketchUp 3D Warehouse [18]. Using COLLADA [19] file format these models were then

imported to Blender for animation and for export to Panda3D in egg format [20]. For a ground model an aerial flat picture was used.

2.2 Implemented visualizations

2.2.1 Excavator position at the site

Excavator working at the site is visualized in 3D. The machine position is moving according the GPS position data received from the machine control system. The test was performed at the Oulu University Construction Automation Test Site. Data was updated every 30 seconds. Excavator was animated to do constant digging movements to please the eye. The program includes an option to change colour to match the current state of the machine, for example red = broken, green = working and orange = on a pause and also colouring the road. At the actual test this was not done however, as the machine control system, at the time of the test, did not support sending status data.



Figure 3. Location visualization

2.2.2 3D objects on the site

There are objects on the road site whose location is important to know when flow of the work is monitored. For visualization freight container as storage for equipment and pile of gravel as buffer storage were included, in Figure 4 blue boxes and brown pile of rocks. These objects can be anything what is decided to be important as adding them is relatively easy task.

2.2.3 4D Timeline visualization - in design phase

This is tool to go through work schedule visually before actual work is done. The idea is to use construction time schedule and visualize in 3D what is planned for the site. This should help the person who makes the schedule to find possible bottlenecks and errors in timetable.

From construction timeline a schedule matrix was created and then used to run visualization. Table shows the components of the matrix. There is one schedule matrix for every individual machine (Machine) which includes start and stop times for planned operation (Task) in selected road layer (Layer) in selected work area (Span)

Table 1. Schedule Matrix – simplified example

Machine 1	Task 1		Task 2	
	Start Span	Stop Layer	Start Span	Stop Layer
Day 1	Start Span	Stop Layer	Start Span	Stop Layer
Day 2	Start Span	Stop Layer	Start Span	Stop Layer

Schedule matrix generation was not automatized, but doing so from project management software was studied and found possible, but was seen unnecessary to this demonstration.

Figure 4 shows a grader driving according to schedule matrix. The green areas of the road layer are ones where the work has been already done and red area is the one where the current work is performed, and white marks layers and areas where work is not done yet. Visualizations were performed to also to excavators (Figure 5) or multiple different machines at the same time as there is no limit how many or what machines are working on the site.



Figure 4. Demonstrating 4D timeline visualization



Figure 5. 4D timeline visualization with two excavators.

3 Results

Extensible Messaging and Presence Protocol was successfully used to transfer position data through internet from the machine control system. This information was then merged to the 3D visualization using open source game engine Panda3D, so that anyone with internet access and the made software, regardless the distance to the work site, can track the position of the excavator.

Because of the work of the manufacturer, Novatron, this task was relatively easy to implement. The machine control system will send the information to the cloud and our task was implement procedures to read the XMPP message using readily available tools, like SleekXMPP library. All main components were using python language making the work even easier.

The negative side of the small open source software was the neglected documentation and possibility to get support right then when needed. On the positive side, support by the users was always there, if sometimes a bit late and the technical skill level of the support was in high level. Main stream open source software like Eclipse and Blender were easy to get support.

Timetable visualization to help to find errors and bottlenecks from the project plan was performed in 3D from the planned work schedule. Graders and excavators were moving on the road work site simulation according to the schedule. Road layers were changing color according the current work phase.

Our work showed that rude 4D simulation using game engine is straight forward engineering work. The object oriented programming made scalability for

different and also for multiple construction machines a somewhat easy task.

Finally a group of 3D objects were added to point the important locations on the site. Freight container demonstrates equipment storage and pile of gravel works as a buffer storage. Adding 3D objects is a relatively easy task as long as one has a 3D model ready for it.

4 Discussion

By opening machine control system data transfer to us to use, in fast phase, we were able to present the position of the excavator real time in 3D visualization running on the open source game engine Panda3D [21]. The data was transferred using Extensible Messaging and Presence Protocol [13] using SleekXMPP [14] library.

Big machine control system manufacturers are keeping the data for themselves, building a business model over the idea that every machine on the site is equipped with their product. This is valid point, but misses the fact that even if the main contractor, in some cases, is fully equipped with only one brand tools the subcontractors are usually not.

If the data is opened for small companies there is a chance that technology will develop really fast. Not only the large manufacturer's software for managing whole constructions digitally would be possible, but there would also be room for smaller companies to fill the caps. At the moment product like Infrakit [3] is working in integration with some machine control systems (Novatron [2], DigPilot [22], Prolec [23]), but is relying on file transfers with other manufacturers. Manual transfers are not real-time, laborious and also leaves room for errors. Gained automation level rise is also not that high when compared to integrated systems.

A rude 4D schedule simulation using open source game engine was performed. From a work schedule a schedule matrix (Table 1) was created, which was then used to drive construction equipment in virtual 3D work site. The virtual site was constructed using layered road model, 3D objects for interesting subjects we wanted to know location for and a flat aerial photo.

The used open source programs were adequate to the job in hand. The grief we had with them, was the not so perfect documentation, which is not so uncommon complain with small software [24].

We suggest that real-time machine control system data should be integrated openly to the building information management process. How exactly one does that is a topic for further study.

References

- [1] "Sitelink3D | Topcon Positioning Systems, Inc." [Online]. Available: <http://www.topconpositioning.com/products/software/network-applications/sitelink3d>. [Accessed: 28-Feb-2014].
- [2] "Novatron Machine Control Systems - Excavator Control Systems - Excavator Guidance Systems - GPS Machine Control - 3D Machine Control." [Online]. Available: <http://www.novatron.fi/en/index.html>. [Accessed: 28-Feb-2014].
- [3] "Infrakit." [Online]. Available: <http://www.hohtolabs.com/en/main>. [Accessed: 01-Mar-2014].
- [4] "SVS Innovations Ltd." [Online]. Available: <http://www.svs.fi/>. [Accessed: 01-Mar-2014].
- [5] "buildingSMART-Tech.org." [Online]. Available: <http://www.buildingsmart-tech.org/>. [Accessed: 01-Mar-2014].
- [6] M. Lu and X. Liang, "Real-Time 3D Positioning and Visualization of Articulated Construction Equipment: Case of Backhoe Excavators," in *Computing in Civil Engineering (2012)*, American Society of Civil Engineers, pp. 196–203.
- [7] V. Kamat and J. Martinez, "Visualizing Simulated Construction Operations in 3D," *J. Comput. Civ. Eng.*, vol. 15, no. 4, pp. 329–337, 2001.
- [8] R. Akhavian and A. Behzadan, "Remote Monitoring of Dynamic Construction Processes Using Automated Equipment Tracking," in *Construction Research Congress 2012*, American Society of Civil Engineers, pp. 1360–1369.
- [9] A. Platt, "4D CAD for highway construction projects, Technical Report No. 54, Computer Integrated Construction Research Program," vol. 2007.
- [10] P. P. A. Zanen, T. Hartmann, S. H. S. Al-Jibouri, and H. W. N. Heijmans, "Using 4D CAD to visualize the impacts of highway construction on the public," *Autom. Constr.*, vol. 32, pp. 136–144, Jul. 2013.
- [11] C. Kim, H. Son, and C. Kim, "Automated construction progress measurement using a 4D building information model and 3D data," *Autom. Constr.*, vol. 31, pp. 75–82, May 2013.
- [12] R. M. Webb and T. C. Haupt, *The potential of 4D CAD as a tool for construction management*. Taylor & Francis, 2003.
- [13] P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies*. O'Reilly Media, Inc., 2009.
- [14] "SleekXMPP · fritzy/SleekXMPP Wiki · GitHub." [Online]. Available: <https://github.com/fritzy/SleekXMPP/wiki>. [Accessed: 01-Mar-2014].
- [15] "Inframodel." [Online]. Available: <http://www.inframodel.fi/>. [Accessed: 27-May-2014].
- [16] D. B. Mathews, *Panda3D 1.6 Game Engine Beginner's Guide*. Packt Publishing Ltd, 2011.
- [17] R. Hess, *Blender Foundations: The Essential Guide to Learning Blender 2.5*. Taylor & Francis, 2013.
- [18] "3D Warehouse." [Online]. Available: <http://sketchup.google.com/3dwarehouse/?hl=en-GB&ct=lc>. [Accessed: 21-Feb-2014].
- [19] "COLLADA.org." [Online]. Available: <https://collada.org/>. [Accessed: 20-Feb-2014].
- [20] "Egg Syntax - Panda3D Manual." [Online]. Available: https://www.panda3d.org/manual/index.php/Egg_Syntax. [Accessed: 21-Feb-2014].
- [21] "Panda3D - Free 3D Game Engine." [Online]. Available: <https://www.panda3d.org/>. [Accessed: 02-Mar-2014].
- [22] "Excavator Guidance | DigPilot." [Online]. Available: <http://www.digpilot.com/>. [Accessed: 02-Mar-2014].
- [23] "Prolec." [Online]. Available: <http://www.prolec.co.uk/>. [Accessed: 02-Mar-2014].
- [24] A. Valtanen, J. J. Ahonen, and P. Savolainen, "Improving the Product Documentation Process of a Small Software Company," in *Product-Focused Software Process Improvement*, Springer, 2009, pp. 303–316.