C. M. Herr, N. Gu, S. Roudavski, M. A. Schnabel, *Circuit Bending, Breaking and Mending: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia,* 219-228.

REAL TIME MODELLING

A solution for accurate, updatable and real-time 3D modelling of as-built architecture

RUSSELL LOWE,[1] JULES CROMARTY[2] and RICHARD GOODWIN[3]
University of New South Wales, Sydney, Australia
1. russell.lowe@unsw.edu.au, 2. j.cromarty@unsw.edu.au,
3. richard@richard-goodwin.com

Abstract. This paper describes a means for 3D modelling that sources photogrammetry data from publically available databases and integrates this data with a real time computer game application to construct point clouds. As the databases receive more data or as the data is updated the 3D visualisation within the computer game environment is capable of remodelling itself to reflect the changes, providing accurate representations of as-built infrastructure within an immersive 3D environment that can be interrogated and analysed in real time.

Keywords. Real-time; photogrammetry; computer game; visualisation; point cloud.

1. Introduction

The range of end users and the resultant need for 3D visualisation applications that suit their requirements has resulted in a fragmentation of 3D modelling and visualisation processes. Many applications exist, some very specific to an end use, and none being a total visualisation solution (if desirable). For the visualisation of as-built architecture, this often results in the need to employ multiple applications to achieve the desired outcome, resulting in complex and tedious workflows and repetition across applications.

Of the various methods to acquire 3D models of existing architecture, the most widely used are CAD tools with 2d architectural plans. Data comprehension remains the longest step in the process with Chevrier identifying 2d plans as the quickest route to understanding the building scheme. (Chevrier and Perrin, 2009) Common issues that run through each modelling tool are the time required to capture or collect accurate and up to date data, the need for manual input to convert this data into 3D models for visualisation and the maintenance of the data to reflect any changes to remain an accurate representation.

The point cloud, a 3D record of points measured from real environments, has emerged as a flexible method for collection and representation of data, with improved potential for updatability due to the relatively fast and automatic capture of data. The combination of high accuracy and abstract representation allows for the flexibility needed at an early design stage whilst maintaining real attributes and the ability to overlay or intersect multiple point clouds enables comparisons between large amounts of 3D data.

Point clouds can be constructed either from laser scanners, light detection and ranging (LiDAR) sensors or photogrammetry input. Photogrammetry has time advantages, brought about by increased community sharing of images and the ability to combine these to form the data set with less set up time (Fumarola and Poelman, 2009). Progress is being made in the modelling of tourist attractions and large city models in applications such as Photosynth which can be accessed here http://photosynth.net/ for public creation of smaller data sets, using personal photos or sourced from the internet. As with other 3D viewing applications, Photosynth suffers from the difficulties of navigating and comprehending large amounts of 3D data.

Computer game technology provides a solution to these difficulties in navigating a virtual model with examples presented in the next section. The main advantage computer game technology has over alternative visualising techniques is the ability for the user to be immersed within the environment and the freedom of movement typical of human interaction within architectural space. While there have been recent developments within CAD applications to this end they are still underdeveloped in comparison with gaming engines. In particular, the presentation of information in real time, that can be experienced by multiple users simultaneously.

Section two of the paper provides an overview of existing research into the use of computer games for visualisation compared to CAD and GIS based 3D visualisation. Current developments of point clouds are discussed and automated modelling from others research is analysed. The literature review identifies gaps within the research that, along with recent advancements within computer gaming engines to provide a new modelling solution, develops the aim of the paper, presented in section three.

The method for this new modelling solution is presented in section four with the introduction of a case study. The results of this case study are presented in section five while section six is an analysis of these results and a discussion on limitations and areas of future development or new applications before the paper concludes in section seven.

2. Background/literature review

Previously identified limitations of CAD and GIS applications for the visualisation of 3D models include difficulties for the average user in seeing space intuitively and in a manner that is easily understood (Germanchis et al, 2005). CAD and GIS initially focused on the 2D view and to a large extent still do, as the 2D view is a common output for work on site. As a result the pan, zoom and rotate tools of 2D documentation have been extended within these applications to provide navigation in 3D as well. Pelosi (2010) acknowledges that while the construction of a 3D model may benefit from these tools, the viewing of a 3D model using these tools can be frustrating. This is not helped by the use of the mouse as the primary method to navigate which Pelosi (2010) identifies as 'a complicated and confusing navigation method to learn and control.'

The popularity of computer games and familiarity of navigation techniques across many users is seen as an advantage over CAD and BIM systems (Pelosi, 2010; Germanchis et al, 2005). However, the intuitiveness of using a keyboard and mouse for navigation within these computer game engines has been questioned (Laing et al, 2007). As with any computer application, computer gaming engines require some getting used to for new users. Should computer game use continue to rise and be more evenly represented throughout the community, as appears to be the trend (Brand-J, 2007), the issue of intuitive navigation for users of computer game engines will decrease. The development of gaming platforms, such as Nintendo Wii and Xbox Kinect, further reduce this issue as navigation is more intuitively linked to user movements. BIM and GIS applications, due to their relatively high costs and specialised use, are unlikely to become more accessible to casual users without a change in their navigation system which is familiar to a wider audience. Such changes have been recently developed for some CAD and GIS applications allowing a first person type walkthrough ability, suggesting that the developers of such applications recognise the benefits that computer game engines have in this area.

There are numerous processes for translating CAD models to computer game engines for improved visualisation capabilities. (Indraprastha and Shinozaki, 2008; Friese et al, 2008; Laing et al, 2007) The time required in all processes is not insignificant and the ability to update changes is limited. Updatability is limited by the need for on-site images to provide texture information and changes in geometry often require editing to the 3D model within the original application before re-exporting to the computer game engine.

Point clouds reduce the need for manual input and can quickly produce detailed 3D models (Shih and Wu, 2005; Fumarola and Poelman, 2009). Nir and Capeluto (2005) identify a growing use of point clouds in architectural design in the later stages of the design or fabrication process. This can be seen in Shih and Wu (2005) where the point cloud is used as a means of measuring as-built progress on a building site. Point clouds are taken once a week and compared by way of intersection to determine the areas of progress. Nir and Capeluto (2005) demonstrate the use of point clouds used for early conceptual design development.

Despite the advantages point clouds have of relatively quick capture and creation of 3D models when compared to CAD and GIS applications they share the difficulty that data is hard to navigate. Perhaps more so; as more data is captured, more points are produced and the difficulty to clearly see the points of interest increases (Chevrier and Perrin, 2009)

3. Aim

The aim of this paper is to present a solution that will provide an end user with access to large, accurate and flexible data within an environment that allows navigation and visualisation amongst multiple simultaneous users. Using computer game engines will enable the data to be more easily interrogated and navigated and challenges the idea that point cloud datasets are not able to be viewed directly within a virtual environment due to their size (Fumarola and Poelman, 2009).

4. Method

The process follows well established data capture and conversion techniques. LiDAR and photogrammetry both yield suitable point cloud data. In the interests of updatability and time saving, photogrammetry shows the most promise. Current community databases contain large amounts of accessible and searchable images and are updated by the community removing the need for manual data collection on the part of the developer.

Photosynth provides the platform for grouping these images together and creates a point cloud as a function of the relationship it creates between the images. Bundler is the basis of Photosynth and if used on its own can create higher detail point clouds but doesn't have the database that Photosynth has of community made point clouds. Using bundler would allow higher resolution images to be used for image matching and hence increase the detail of a point cloud. However, this would limit the automation of the workflow as point clouds would always need to be created, whereas Photosynth has an existing library of point clouds already converted from the raw images. It is quite likely, as this database grows, locations that a user may be looking to document have already been uploaded to Photosynth, and as these locations change, users may produce newer point clouds that reflect these changes.

Point clouds are exported from Photosynth using an open source exporter, SynthExporter, available at http://synthexport.codeplex.com. The exported file created is a *.ply file and is readable and editable in any text application. Data captured from LiDAR would remove this step as the point cloud would have already been captured. However, additional steps through 3D modelling applications would be required to achieve a suitable file format to link with the computer game environment.

The computer gaming engine used is the CryEngine2 gaming engine developed by Crytek and used to develop the Crysis series of first person shooters. Previous research (Friese et al, 2008) identifies advantages that CryEngine2 has over alternatives, namely CryEngine's editing tools.

The link between the point cloud data and the computer game environment is achieved through the application of a Flowgraph Plugin System (FGPS) developed by a member of the Crysis community, James Ryan (www.crymod.com) for CryEngine2. The FGPS allows the Sandbox2 to interrogate *.xml files so that they may influence a variety of game play elements in real time. The *.ply files are converted to *.xml in a format that represents each vertex and associated colour information in consecutive nodes. The FGPS reads through each node and responds to any changes as they occur. These changes are then viewed and interacted with as the user moves around within an immersive 3D environment. Users are able to view the data from any perspective but also influence the data itself.

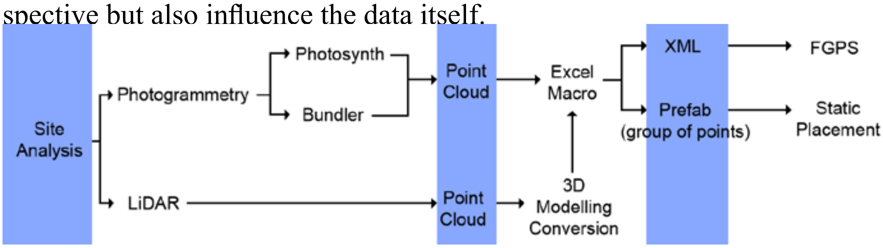

Figure 1. Diagram of method.

The workflow in Figure 1 is demonstrated in an Australian Research Council funded project (ARC LP 0991589) that is contributing to our understanding of urban space in a major Australian city. The project involves crowd modelling and analysis utilising sensor driven avatars within a computer game environment (CAADRIA 2011, Lowe, Hedley and Goodwin). In the case of this crowd modelling and analysis an abstract point cloud model was not a desired end result, rather a typical 3D model was required for user recognition for visualisation and simulation exercises.

5. Results

A number of point clouds were successfully created within the gaming environment, having been sourced from various point cloud creation techniques. Initially small sample sizes of point clouds that were manually created within a text application were brought into the game environment to test the workflow and the computer games ability to manage the data. Ply files were then exported from various existing Photosynth files available in the community. An example is shown in Figure 2. These point clouds require some editing to get them into an *.xml format that the FGPS can read. This is done using an Excel spreadsheet and a macro script created for this purpose. This proved the

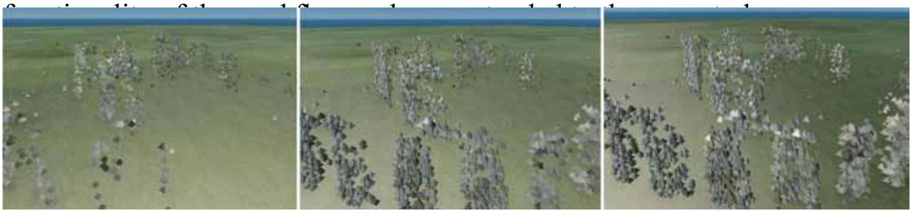

Figure 2. Stonehenge Photosynth as point cloud within gaming engine showing increasing detail – 1500, 3000, 4500 points.

5.1. CASE STUDY

To produce a 3D model that was recognisable for simulation exercises was problematic due to the buildings age, lack of historical importance and consequent lack of accurate architectural drawings as well as the fragmentation of these drawings. In this case study the point cloud creation and integration within the computer game provided a technique for checking the accuracy of the model constructed from drawings. The point cloud was constructed using Photosynth and images extracted from video footage taken during site visits.

This represents a likely process in the case of specific projects where data is not readily available and community image databases do not have the necessary amount of information. This could occur in areas that are not popular tourist destinations or where photography is limited for security or other reasons. The decision to capture site information using a video camera made data collection less intrusive in an area continuously being used by the public. Video footage also ensured there was adequate coverage of the area, whereas capturing static photos was more likely to have missed areas of interest and possibly requiring more time consuming site visits.

The point cloud data was then compared to the 3D constructed model to check the 3D constructed models accuracy. Such a use for point clouds has been used extensively in previous studies and applications (Shih and Wu, 2005; Fumarola and Poelman, 2009). The added advantage that this process has, is the ability to combine the two models within the computer game environment. This allows multiple users to analyse the two models from any perspective, independent of each other. It produces a virtual site visit in which discussions can occur within a group as each member interrogates the model. Figure 3 shows views typical of what a user would encounter within the game environm … el.

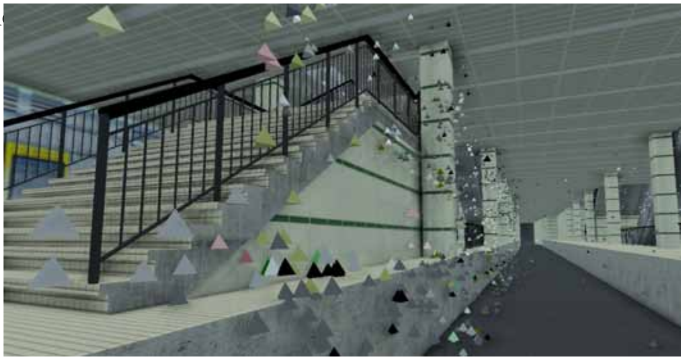

Figure 3. Conventional 3D model with overlaid point cloud within gaming engine.

This process, from Photosynth to CryEngine maintains colour information from the *.ply file and allows each individual point to be created within the gaming environment with its correct RGB colour originally captured by the photogrammetry. This removes the need for texture creation or mapping within a 3D modelling program, a process that is required for almost all visualisation applications and one that has been regularly identified as requiring a lot of time and manual input (Shih and Wu, 2005; Laing et al, 2007; Fumarola and Poelman, 2009). In addition there is little scope to maintain this texture data in a typical application as any changes will require the modeller to start from the data capture step in order to update the texture information.

It is possible to import the point cloud data into the game in a number of ways. One option was for each point to be created (the FGPS refers to this as the point being “spawned”) individually, one after the other. The rate at which points are spawned and how many are spawned, or the interval between each being spawned, can be controlled and allows for a flexible form of analysis. Alternatively a group of points can be spawned together to get a more immediate representation of the point cloud but also to break up the point cloud for visual, functional or analytical reasons.

Currently points are spawned based on a default order that is generated in the conversion from Photosynth to the *.ply file. As a result each point is identified based on this order. However the FGPS allows for great flexibility in this area with scope for more specific identifiers for editing and change management. The FGPS also allows for the export of the point cloud information from the gaming environment which provides a robust system to compare and analyse changes.

6. Discussion and further opportunities/applications

The ability to visualise 3D point clouds within an immersive computer game environment has obvious advantages for viewing information and being able to navigate it successfully (Manovich, 2000). Typical mesh and texture mapped 3D models are unlikely to be replaced by this workflow in the short to medium term as they still represent the most accessible, conventional and photo realistic environments to the public (Fumarola and Poelman, 2009).

From the analysis of the case study and the previous trials of manually created point clouds it is evident that the combination of current computer memory allocation and the generation of point clouds as massive collections of simple objects (four polygon tetrahedrons) limits the size of the point cloud representation in Crysis. On the computers used for testing (high powered graphics workstations), point clouds larger than 10,000 flashed and became unstable graphically when imported as a group. Similarly, individual points spawned from a point cloud of around 10,000 suffered some time delay with significant drop in frame rates as the computer game scene became over populated. It is expected that computer speeds and memory limits will improve significantly, thereby improving the scope of these results. There is however a large gap between 10,000 points and the million or so points one regularly see’s in point clouds of popular environments.

One branch of our future work in this area involves automatic meshing of point cloud data captured by robotic devices and initial experiments show enormous promise where Crysis represents 250 meshes (instead of 10,000) but with each mesh containing 10,000 polygons, with no noticeable issues.

Another area of exploration being pursued is the ability to create meshes within the game instead of geometry to represent the position of the points. It is feasible that instead of spawning geometry at the x, y, z locations contained within the *.ply file that a flowgraph could be developed to connect vertices in the point cloud and create polygons creating a 3D mesh in the computer game environment. Conversely, if computer memory and application limitations allowed, point clouds could be made dense enough so as to remove the need for mesh geometry. Ultimately these approaches could replace 3D modelling software in creating photorealistic virtual environments.

Finally, an extension of the workflow is being explored to provide a system for modular and prefab design. As the point cloud data determines where a mesh is spawned within the computer game and the user has control over what meshes are spawned at each point it is possible to create a matrix of positions of regular elements. This matrix can then be edited, just as the unstructured point clouds obtained from photogrammetry can be. A design can then be modelled from a library of parts and edited quickly and accurately. The design of a health care facility is an example of an architectural program that would benefit from this approach as the spatial requirements are quite rigid with rooms and equipment repeated regularly.

There remain some gaps in the automation of the workflow that require manual input. The process of exporting a *.ply from Photosynth and converting it to a suitable *.xml format is an example of one of these gaps. The development of the macro script mentioned above greatly improved this but should be taken further. An application will be created that can continually export Photosynth point clouds and reformat them in a FGPS readable format to keep the content as current as possible, if not in real-time.

7. Conclusion

The workflow presented above can immediately benefit the development of conceptual designs. Environments can be created in which conceptual designs can be experienced, interrogated and edited in real time through the use of the FGPS, while being located in a spatially, geometrically and texturally realistic representation of existing surrounding architecture and landscape. This extends existing applications of point clouds for conceptual design but by including site information earlier can improve the design decisions at an earlier stage (Nir and Capeluto, 2005).

The creation of a 3D model for typical visualisation purposes is time consuming and is often repeated by multiple modellers for the same building. Checking the accuracy of such 3D models against existing environmental conditions can also be as time consuming and as tedious as creating a new 3D model. The point cloud workflow may be an alternative to multiple site visits and can remove the need for site visits all together if community image databases contain suitable information for the project. Coupled with the ability to easily navigate the point cloud within an immersive computer game environment, the workflow demonstrated provides a viable solution to providing reliable 3D models. The accuracy of the point clouds within the 3D environment was validated by a comparison to a 3D model created using accurate architectural and engineering drawings as can be seen in Figure 3. The workflow may also greatly reduce the time and manual input of existing processes whilst adding confidence in the accuracy of any conventionally created models that may still be useful.

Acknowledgements

The research is jointly funded by the Australian Research Council (ARC LP 0991589), the University of New South Wales (UNSW), and the Emergency Information Coordination Unit (EICU).

References

Al-Douri, F. A., Clayton, M. J. and Abrams, R. F.: 2005, The impact of 3D digital modelling on the 3D design aspects in urban design plans, *eCAADe 23*, Lisbon, 331-340.

Brand, J.: 2007, “Interactive Australia 2007: facts about the Australian computer and video game industry”. Available from: Open Source Repository <http://epublications.bond.edu.au/hss_pubs/95/> (accessed 26 November 2010).

Chevrier, C. and Perrin, J. P.: 2009, Generation of architectural parametric components, *CAAD-Futures 2009*, Montreal, 105-118.

Friese, K-I., Herrlich, M. and Wolter F-E.: 2008, Using game engines for visualisation in scientific applications, in P. Ciancarini, R. Nakatsu, M. Rauterberg and M. Roccetti (eds.), New frontiers for entertainment computing, Springer-Boston, Milano.

Fumarola, M. and Poelman, R.: 2009, Case studies on the generation of virtual environments of real world facilities, *CONVR 9*, Sydney, 155-164.

Germanchis, T., Cartwright, W. and Pettit, C.: 2005, Using computer technology to explore human wayfinding and navigation abilities within a built environment, ICC 21, Coruna.

Indraprastha, A. and Shinozaki, M.: 2008, Constructing virtual urban environment using game technology, *eCAADe 26*, Antwerpen, 359-366.

Kobayashi, Y., Grasso, C. J. and McDearmon, M. J.: 2010, World 16: innovation and collaboration in VR technology, *eCAADe 28,* Zurich, 593-603.

Laing, R., Conniff, A., Craig, T., Diaz, C. G., and Scott, S.: 2007, Design and use of a virtual heritage model to enable a comparison of active navigation of buildings and spaces with passive observation, *Automation in construction*, **16**(6), 830-841.

Manovich, L. 2000. The Language of New Media. Cambridge, MA: MIT Press.

Nir, E. and Capeluto, G.: 2005, Smart cloud-of-points model, *eCAADe 23*, Lisbon, 687-694.

Pelosi, A.: 2010, Obstacles of utilising real-time 3D visualisation in architectural representation and documentation, CAADRIA2010, Hong Kong, 391–400.

Shih, N. J. and Wu, M. C.: 2005, A 3D point-cloud based verification of as-built construction progress, *CAADFutures 2005,* Vienna, 193-202.