

3DSYSTEK Web-based Point Cloud Viewer

E. Maravelakis, A. Konstantaras

School of Applied Sciences
Technological Educational Institute of Crete
Chania, Greece

Corresponding Author: marvel@chania.teicrete.gr

K. Kabassi

Dpt of Protection and Conservation of Cultural Heritage
Technological Educational Institute of the Ionian Islands
Zakynthos, Greece

I. Chrysakis, C. Georgis, A. Axaridou

Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
Heraklion, Greece

Abstract—This paper presents the development of a new Web based LiDAR (Light Detection And Ranging) 3D point cloud viewer addressing mobility and portability issues arising from remote field applications by numerous multidisciplinary collaborative scientists. This new Web Browser-based 3D point cloud viewer, hereafter called 3DSYSTEK viewer, was developed and implemented in the framework of the 3DSYSTEK research programme that aimed to develop tools to bring together all disciplines working to promote our cultural heritage through laser scanning, 3D modeling, 3D visualization, documentation, and exploitation of 3D data. 3D point cloud viewers' native implementations although they support multiple applications they are very expensive and their features vary from one software to another with considerable problems arising due to the introduction of personalized file formats by each individual native software. The presented 3DSYSTEK viewer is based on open-software, enables personalized expandability to address specific needs by individual users and allows the online remote collaboration amongst scientists at different locations as well as the widespread usage of the TLS surveying technology and 3D modeling of large monuments.

Keywords—3D point cloud; Terrestrial Laser Scanners, web browser based 3D point cloud viewer; web graphics library; particle system technology

I. INTRODUCTION

Terrestrial Laser Scanner (TLS) technologies enabled the near instant gathering of a vast volume of 3D point measurements, including data for XYZ coordinates, color information and orientation in space. The first product of a TLS is the point cloud which is processed in the next stage, usually leading to a mesh model, according to user needs and the required deliverables [1]. For this reason a remote, user friendly, no plug-in required, point cloud visualization tool is very important for providing a first web-based deliverable which can be also used for a quick assessment of the data quality and the planning of the next processing phase [2].

A literature review indicates the presence of several web-based 3d mesh model viewers, whilst there are very few 3D point cloud web-based viewers [3]. These are mainly commercial 3D viewers produced by TLS technology providers companies, eg. Faro Scene Web Share, Leica Tru View, Allpoint Insight, Point Cab web Export, etc. Their software is heavily focused towards mapping particular TLS hardware capabilities to key features supported by its sister customized 3D viewer. As a result several problems arise in terms of multiple data types, features' limitations, expandability issues, requirement of plug-ins and data extraction functions. Also there are few non-commercial web-based point cloud viewers, which are mainly experimental, demonstrating limitations encountered due to large storage requirements, memory management deficiencies, lesser transportability due to file sizes and real time visualization [4]. Furthermore, there are no web-based 3D point cloud viewers offered as products in the market that match by any means the capabilities of those in native implementations.

This paper presents a novel web-based 3D point cloud viewer, based on open software and open web technologies, easily expandable, without any plugin or installation requirements, supporting the most common XYZRGB format, which makes it compatible to all types of TLS. The proposed 3D point cloud viewer is based upon particle system technology [5], that enables the entire point cloud to be treated as a whole object, and thus allow for the effective handling of large point clouds.

II. 3DSYSTEK VIEWER

The proposed 3DSYSTEK viewer aims to address these issues by being implemented as a web-browser application based on open software. For its development we have used an open source WebGL Abstraction Library called ThreeJS[6] while the code is written in Javascript using the open web technologies HTML5 and CSS3 to implement the user interface. WebGL is a Javascript application programming interface (API) based on the popular OpenGL API that

provides 3D rendering capabilities from within a Web Browser. The OpenGL API has a long history in 3D Graphics, having been used for all kinds of 3D applications since its first version in 1991 and is currently one of the most supported 3D APIs available. WebGL, being relatively new, is not supported by older browsers or by older hardware compared to the longer-lived OpenGL which is provided as native software for each platform.

The advantages from a 3D point cloud viewer on a web-browser, such as 3DSYSTEMEK viewer are multiple. It can run anywhere with no need for download. It is a ‘write once - run anywhere’ application with increased portability as it can run on Windows, Linux or MacOSX operating systems, mobile phones, tablets, etc. The fact that it can be used remotely and by several types of electronic devices supports the collaboration amongst teams of scientists in remote locations as it can be used equally as easy both on sight and in laboratory applications. Furthermore, the use of open source technologies allows the expandability of the existing software by anyone to match specific needs of individual research groups. All that is required, is for other developing teams to download the software source code and then they are free to expand on it any way they see fit.

No new technology, though, comes without down faults, and it is the intention of the authors not to withhold any significant information that may impose any difficulties to the users. WebGL technology is not yet entirely standardized. It is still under development which means that some of its aspects may change, i.e. it is not entirely future proof, and software written today may not operate on future versions. Because it is web-based it is slower than native implementations but this is an issue that will only get better in time as it is under heavy development by multiple organizations and due to the current trend for new hardware to better support open graphics libraries, such as WebGL, OpenGL, etc. Memory management issues and speed limitations prohibit the handling of very large data sets that are better suited for native implementations. The current capabilities of 3DSYSTEMEK viewer just exceed ten million points in any particular point cloud. Yet again, the newly introduced web-workers technology [7], similar to threaded applications, will alleviate some of the problems in speed and scalability of datasets.

The basic philosophy behind the 3DSYSTEMEK viewer is that we use particle system technology [5] for speed and ease of handling of the entire point cloud as a whole object. This means that the particle system moves all points in the cloud as a unified object instead of having to continuously compensate for the motion of all other points in relation to the motion of any one point. Currently we are using perspective projection of the 3D scene from one camera to render the 3D scene. Future implementations will encompass the use of additional orthographic cameras to display simultaneously a point cloud from different angles. This will provide a better spatial impression and improved alignment capabilities of the point cloud in relation to world coordinates/world plane. In 3DSYSTEMEK viewer when the user moves the camera its coordinates are automatically transformed in order to ensure that the point cloud, which is our object of interest, is always contained in the view frustum [8]. In explaining, in 3D

computer graphics the viewing frustum is the region of space in the modeled world that may appear on the screen, thus by varying the camera coordinates we ensure that the object of interest always falls within the field of view of the notional camera.

III. OPERATION ANALYSIS

The following schematics display the asynchronous operations initiated by the 3DSYSTEMEK viewer template relating user actions and software events with software actions and their corresponding outcome.

A. 3DSYSTEMEK template operations

The user can control the movement and position of the notional camera either by using the mouse or by importing values in the control field parameters. The latter is dynamic and always displays the values corresponding to the instant orientation of the camera. These are shown in Fig. 1.

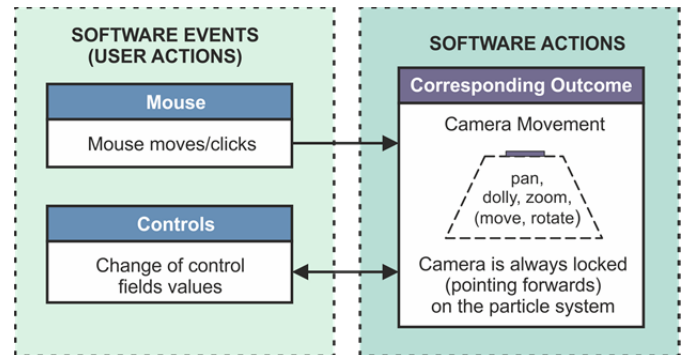


Figure 1: 3DSYSTEMEK manual controls

Another field is used to input a 3D data file to the 3DSYSTEMEK viewer. Upon insertion of an input data file a chain of operations is being initiated. Firstly, application programming interface (API) is employed to open the 3D data file from the local file system to the HTML5 browser followed by 3D data parsing to create a particle system from the imported XYZRGB points in the input file and then the particle system is added in the 3D scene. A reset option is also available through a dynamic button which resets the notional camera view to the one emerging from its rest coordinates. These are outlined in Fig. 2.

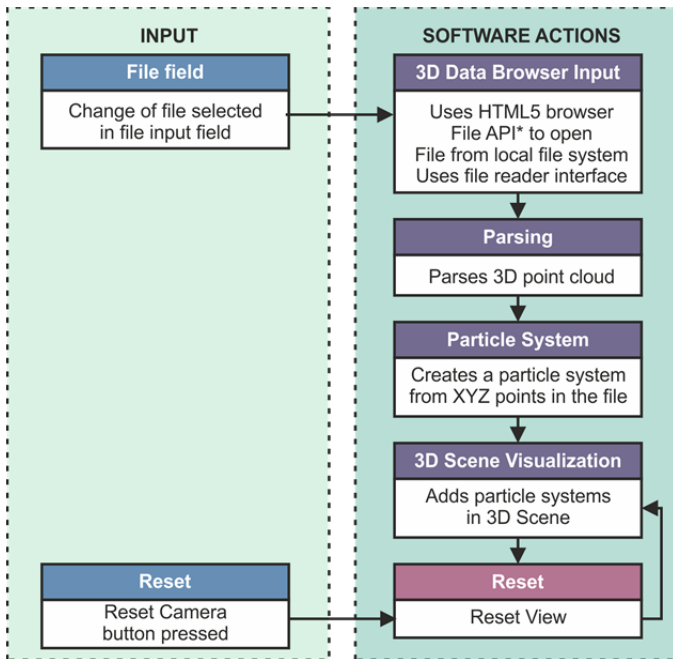


Figure 2: 3DSYSTEMK point cloud management system

B. Point cloud and image alignment

A key feature of 3DSYSTEMK viewer apart from displaying color 3D images at various angles and different zooms is the capability to align point clouds to 2D images imported to the 3D point cloud viewer from the external world. This feature finds significant applications in archaeological sites' modeling, city modeling, 3D model projections to existing 2D maps and features extraction, etc [9]. In this key feature of 3DSYSTEMK viewer the point cloud is visualized using WebGL while the 2D image is displayed using traditional browser HTML/CSS technologies. The point cloud is contained in a HTML5 3D canvas object[10]. In explaining, WebGL is based on OpenGL ES 2.0 and provides an API for 3D graphics, it uses the HTML5 canvas element and is accessed using Document Object Model interfaces, while automatic memory management is provided as part of the JavaScript language. The image is used as a background of a DOM block element [11]. DOM, short for Document Object Model is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents within the Web browser. Both the canvas and the DOM block element are partially transparent using their CSS opacity properties. CSS stands for Cascading Style Sheets [12] and is a style sheet language used for describing the look and formatting of a document written in a markup language (in this case HTML). We use this for the application user interface and the displaying of the image that will be aligned with the point cloud (Fig. 3).

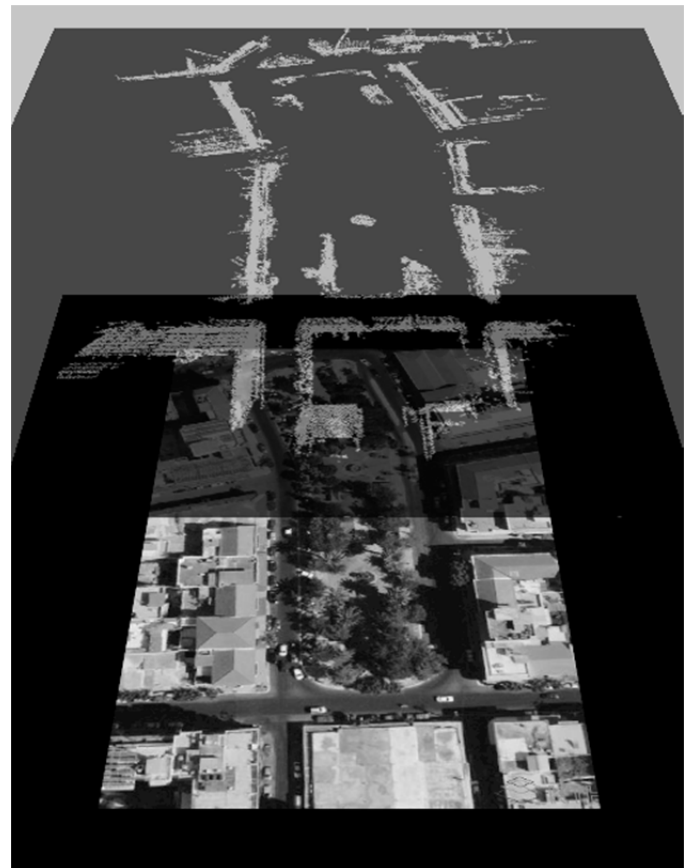


Figure 3.3D point cloud aligned against its 2D aerial orthophoto

This is done to simplify and speed up the development and update process. The alternative approach would be to use the image as a texture of a 3D plane within the 3D Canvas object, but that would require constant complex calculations whenever the camera was moved, in order to always keep the plane position static in relation to the user's view. Since we only require the relative scale of the image and the rotation/position of the point cloud in relation to the image, there is no reason to involve the image as a 3D object. The core advantages of the proposed approach are better performance, flexible development and upgrading.

The image alignment (Fig.4) could be a simplified process by using the 3DSYSTEMK viewer. As shown in the asynchronous operations of the above figure, all the user has to do is importing the 2D image to the 3DSYSTEMK viewer which is displayed by the viewer as background to the DOM element of the browser. Then the user can use the image controls to change the scale input value which updates the image DOM element background dimensions and alignment using CSS properties. There is also a reset dynamic button that returns the background dimensions and alignment to their original values. When the user is happy with the alignment he/she can lock the point cloud to the 2D image, saving the bond and not having to repeat the process every time he/she wishes to align the two together.

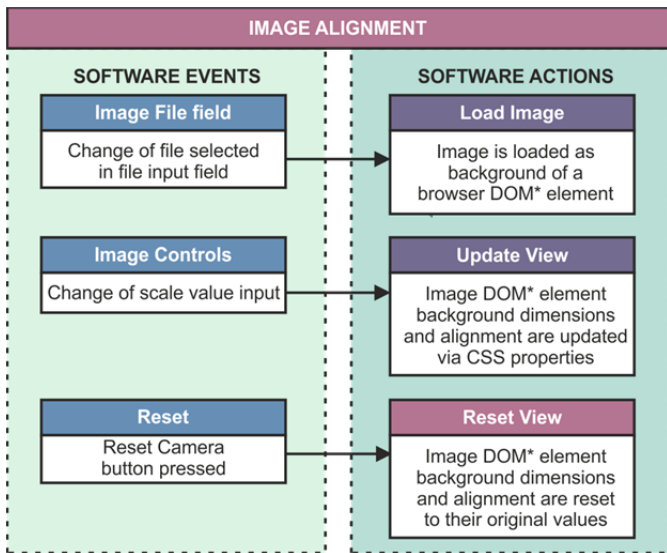


Figure 4. Point cloud and image alignment system

IV. DEMONSTRATION RESULTS AND CONCLUSIONS

The following results demonstrate the capabilities of 3DSYSTEMEK viewer in various popular applications each highlighting certain features of the software. The first one shown in Fig.5 is a 3D colored point cloud of cultural exhibit of medium complexity. This object was selected to highlight the capability of the 3D viewer in color management and its display features in a compact object. The point cloud was obtained by a Minolta Vivid 910 3D laser scanner. This application is becoming more and more popular with digital museums and with the archeological society as it provides a non-destructive documentation and electronic replication of ancient artifacts.



Figure 5. 3D colored point cloud of cultural exhibit

Fig.6 is a false color 3D point cloud of the monumental olive tree of Vouves, known as the oldest olive tree of the world [13]. The object is of the outmost complexity in terms of shape, size variations, detailed features and asymmetry from any point of view. As seen in the figure the 3DSYSTEMEK viewer was capable to depict, handle and present clearly and in great detail the point cloud imported to it. Future work of the authors could include the incorporation of annotation capabilities and documentation on certain parts of complex objects.

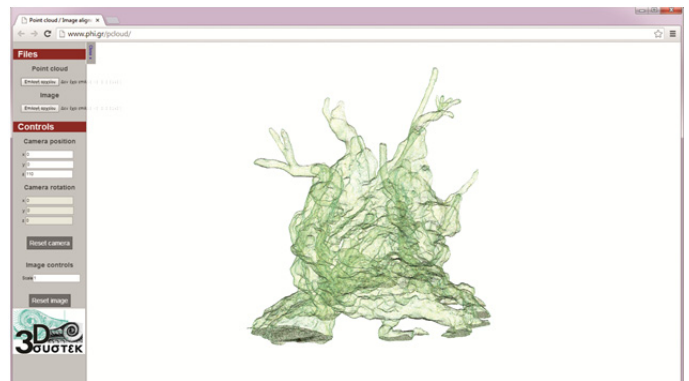


Figure 6. 3D point cloud of the monumental olive tree of Vouves

The 3D point cloud model of the ancient castle of Aptera, consisting of 9,5 million points was selected to highlight the ability of 3DSYSTEMEK viewer to present large scale objects (Fig. 7). The web browser based 3DSYSTEMEK viewer can comfortably manage up to ten million points in a point cloud and we are actively working towards doubling that number. This application finds great use in modeling wide cultural, urban and rural areas of interest, enhancing the documentation of monuments, land degradation, cities expansion, etc. [14]. The 3D point cloud is often merged with aerial photography to extract 3D information of the object(s) of interest. That was the steam idea that led us towards developing and incorporating the point-cloud & image alignment tool in our 3D web browser-based point cloud viewer.



Figure 7. 3D point cloud model of the ancient castle of Aptera

The above demonstrated capabilities of the 3DSYSTEMEK point cloud viewer that will allow the widespread usage of the TLS surveying technology and 3D modeling of large monuments. This is based on the maximum exploitation of the radical innovative advantages of TLS, enhanced by key features of the host Browser-Tool [15-16]. This is devoted to searching, querying, browsing, and viewing multimedia objects stored in the repository, based on the Fundamental Categories of 3D objects, (Actors, Things, Events, Place, Time, Type) which are connected by using interpretations of "Fundamental Relationships" (refers to, is from, has part, is similar, has met, at), but also on keyword search.

ACKNOWLEDGMENT

The authors wish to thank the General Secretariat for Research and Technology of Ministry of Education and Religious Affairs, Culture and Sports in Greece for their financial support (via program Cooperation: Partnership of Production and Research Institutions in Small and Medium Scale Projects, Project Title: “3D-SYSTEK - Development of a novel system for 3D Documentation, Promotion, and Exploitation of Cultural Heritage Monuments via 3D data acquisition, 3D modeling and metadata recording”).

REFERENCES

- [1] E. Maravelakis, A. Konstantaras, A. Kritsotaki, D. Angelakis, M. Xinogalos, “Analysing User Needs for a Unified 3D Metadata Recording and Exploitation of Cultural Heritage Monuments System”, *Advances in Visual Computing, Lecture Notes in Computer Science*, Volume 8034, 2013, pp 138-147.
- [2] J. Fernandez, A. Singhanian, J. Caceres, K. Slatton, and R. K. M. Starek, “An overview of lidar processing software Technical report”, Geosensing Engineering and Mapping, Civil and Coastal Engineering Department, University of Florida, 2007.
- [3] M. De La Calle, D. Gómez-Deck, O. Koehler and F. Pulido, “Point cloud visualization in an open source 3d glob3”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXVIII-5/W16, 2011 ISPRS Trento 2011 Workshop, 2-4 March 2011, Trento, Italy
- [4] M. Kuder and B. Zalik, “Web-based LiDAR Visualization with Point-based Rendering”, *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*, 978-0-7695-4635-3/11 © 2011 IEEE DOI 10.1109/SITIS.2011.36
- [5] W. Reeves, “Particle Systems: A Technique for Modeling a Class of Fuzzy Objects”, *ACM Transactions on Graphics*, April 1983
- [6] J. Dirksen, “Learning Three.js: The JavaScript 3D Library for WebGL”, ISBN 1782166289, 2013
- [7] HTML Standard – Web Workers [online], <http://www.whatwg.org/specs/web-apps/current-work/multipage/workers.html#workers>, 2014
- [8] Microsoft Developer Network [online], <http://msdn.microsoft.com/en-us/library/ff634570.aspx>, 2014
- [9] S. Sun and C. Salvaggio, “Aerial 3D Building Detection and Modeling From Airborne LiDAR Point Clouds”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Volume 6 (3), 2013, pp 1440-1449
- [10] S. Fulton and J. Fulton, “HTML5 Canvas”, O’Reilly Media, 2011
- [11] J. Marini, “Document Object Model : Processing Structured Documents”, McGraw-Hill/Osborne, 2002
- [12] E. Meyer, “CSS: The Definitive Guide”, O’Reilly, 2007
- [13] E. Maravelakis, N. Bilalis, I. Mantzorou, A. Konstantaras and A. Antoniadis, “3D modelling of the oldest olive tree of the world”, *IJGER*, Volume 2(2), 2012, pp. 340-347
- [14] D. Abate, S. Migliori, S. Pierattini, B. Fernández-Palacios, A. Rizzi and F. Remondino, “Remote rendering and visualization of large textured 3D models”, *Proceedings of the 2012 18th International Conference on Virtual Systems and Multimedia, VSMM 2012: Virtual Systems in the Information Society*, art. no. 6365951, 978-1-4673-2565-3/12/©2012, IEEE, 2012, pp. 399-404
- [15] M. Doerr, I. Chrysakis, A. Axaridou, M. Theodoridou, C. Georgis and E. Maravelakis, “A framework for maintaining provenance information of cultural heritage 3D-models”, *EVA London*, 2014, pp 267-274.
- [16] K. Maragkogiannis, D. Kolokotsa, E. Maravelakis, A. Konstantaras, Combining terrestrial laser scanning and computational fluid dynamics for the study of the urban thermal environment, *Sustainable Cities and Society*, Volume 13, October 2014, pp 207-216