COMBINED 2D AND 3D WEB-BASED VISUALISATION OF ON-SET BIG MEDIA DATA

Alun Evans *Javi Agenjo* *Josep Blat*
GTI (Interactive Technologies Group), Universitat Pompeu Fabra, Barcelona, Spain
{alun.evans, javier.agenjo, josep.blat}@upf.edu

ABSTRACT

In this paper, we present a web-based application allowing the analytic visualisation of on-set media data and metadata, which combines research from several fields of image processing and 3D graphics. Film and television production is currently undergoing a paradigm shift, away from the concept of capturing a 'shot' with a single principal camera view, to recording everything possible about the set, the environment, the objects and the actors within it. In parallel, the rise of cloud-based storage and processing is pushing towards applications being controlled via web or browser based interfaces. The application presented in this paper has been created to meet both of these needs, visualising Big Media Data via a web-based interface, and supporting the integration of multiple data formats and analysis. We discuss the different modalities and processing included in the current tool, stress the integration into the web and the solutions to the challenges posed, and show how the tool has been initially tested by professionals, with positive outcomes.

***Index Terms*—**Big Data, 3D, Application, WebGL

1. INTRODUCTION

Film and television production is currently undergoing a paradigm shift, away from the concept of capturing a 'shot' with a single principal camera view, to recording everything possible about the set, the environment, the objects and the actors within it. The interplay between production, pre- and post-production is changing too, with boundaries between each phase blurring or even disappearing. In parallel, the rise of cloud-based storage and processing is pushing towards applications being controlled via web or browser based interfaces. In this paper, we present a web-based application allowing the analytic visualisation of on-set media data and metadata, which combines research from several fields of image processing and 3D graphics. The data might come from multicamera setups, spherical images or LIDAR (3D scans), whereas the metadata can be 3D reconstruction of the set, registration, action recognition, saliency and coverage of the cameras. Through a web-based 3D viewer, and the assistance of a timeline interface, camera footage and associated metadata are viewed in the context of the LIDAR data, considered as a ground truth 3D environment.

Ideally, big data visualization should support integration of data from multiple sources, automated analysis and user decisions. The variety and size of on-set recorded data is increasing hugely (6TB are currently generated each shooting day, from GPS, optical or other sensors, spherical, witness, principal cameras etc.). Each modality of data is visualized and analyzed separately, and any integration requires a lot of human resources. The analysis and integration usually take place in post-production stages, away from the on-set, production shooting. If quality issues are identified, the worst case is that footage will have to be re-shot, which is slow and extremely expensive; in the best case, issues can be fixed through post-processing, which is very costly. Furthermore, visualizing each modality on its own is a slow process; users have difficulties to understand the whole picture, and to identify the quality issues. The paradigm on which the tools presented in this paper is based is to provide an integrated view of the different modalities of data, with some processing/metadata incorporated early in the process, so that the analysis can be made more easily. The actual shooting environment is simulated, efficiently allowing decisions to be taken near the on-set situation, either real-time, near real time, or before the next day shooting. In the tools presented in the paper, video captured through multiple cameras is shown from the appropriate position and orientation in a 3D on-set context constituted by LIDAR scanned data, implicitly using the 3D reconstruction and registration metadata. Other metadata used in the visualization are recognized actor actions, along with the most salient camera for those actions. Quality issues are coverage of the scene, and the saliency itself.

Why web based tools? Users increasingly need to access applications from anywhere, and on any platform, and need to be able to share and collaborate. This includes both those professionals present on-set (a large crew with different roles, generating different modalities of data and metadata), and those remote, who have roles in the production and post-production processes, visual effects, etc. There is a strong drive for any on-set visualisation app to be web-based, as this requires no external software to be installed and, by its very nature, is suited for remote access to the data, and to support collaboration. On the other hand, combinations of 3D and other modalities on the web have recently appeared, as shown by Jankowski et al. [1], [2], who presented an interface combining hypertext and simple 3D graphics, and showed that the performance with this so called “dual-mode” interface was better than for single modalities (even taking into account switches). The visualization we present is more advanced in several respects, in that it combines more challenging 3D data, layered 2D data and metadata, and it has much higher user interactivity than Jankowski et al. The combination of modalities, the efficient use of bandwidth and the processing at the client side, which has implications on usability, pose significant challenges.

In the rest of the paper we discuss the different modalities and processing included in the current tool, with a stress on the integration into the web, and the solutions to the challenges posed. We discuss the positive outcomes of the initial tests with professionals, and the extensions envisaged towards. a fully operational tool for on-set work.

2. WEBGL AND 3D ON THE WEB

Web-based 3D graphics has seen dramatic progress in recent years, due to the release of the WebGL standard in 2011. Now supported by all major desktop and mobile browsers, the WebGL API allows the browser to access hardware accelerated graphics without requiring 3rd party plugins such as Unity3D [3] or Adobe Flash [4]. WebGL can be programmed imperatively directly via the browser using Javascript, although several more declarative methods of programming Web-based 3D have gained popularity in the research domain [5], [6]. Web-based 3D applications share many of the advantages common to all web-based technology, namely platform independence, no reliance on 3rd party software, and ease of distribution and maintenance. Using WebGL also allows a seamless integration of 2D and 3D which facilitates the creation of powerful and innovative interfaces [2], [7], which would be more difficult to create with non-browser-based software. However, the difficulties inherent to many client-server based technologies, such as those relating to bandwidth and synchronisation, are particularly present in all web 3D applications due to the typically large file sizes of the assets used. For more information on the current state of the art in web-based 3D graphics, we refer the reader to a recent comprehensive survey [8].

2.1. Progressive Transmission of Pointcloud Data

Light Detection and Ranging (LIDAR) is a remote sensing technique that uses the time difference between the transmission of a laser pulse and the detection of its reflection to calculate the position of an observed point on the surface of an object. The information gathered from the pulses is then amalgamated and converted into a topological cloud of points in 3D space, which can be stored on disk in one of several formats (for example, LAS or OFF). LIDAR machines will be set up on-set to record the filming environment, which can then be viewed later to assist in post-production, whether as a visual aid to assist in the creation and/or lighting of digital assets, or a ground truth to assess registration techniques involving multiple cameras, or simply to provide better context of the set to the post-production staff. Although laser scanning can generate highly accurate point clouds of a surface, the resulting data can consist of millions of 3D points (and associated colour information). The large file sizes involved present considerable problems in terms of the storage, transmission and visualization of the data, and there has been much previous research on issues of compression and progressive visualization [9]–[14]. However, to-date there has been very little work focusing on the progressive transfer of point-cloud data for remote visualisation via the internet [15].

The problems facing any system of remote visualisation of big media data are twofold: firstly, typical bandwidth capabilities result in unacceptable waiting times if the user wishes to visualise the entire data set; and secondly, the comparative lack of computing power of javascript-based processing within the browser context means that heavy compression techniques may be take too long to decompress the data and be counter productive [16].

For this work, we use an approach similar to that of [15] to progressively visualise point cloud data in a WebGL content. The goal is for the 3D context to quickly display a low-resolution version of the data, which then is progressively refined to higher resolution as more data is downloaded. To do this, an offline process is used initially to store the point cloud data into a memory efficient octree data structure. Higher level octree data (i.e. smaller in size) is sent from the server to the client and displayed in the 3D context with little delay, and the display is updated as information about lower levels of the octree is downloaded.

Once the octree has been created, it is then traversed in a breadth-first manner and relevant information for each node is stored to file using a custom binary data format is used to store the relevant information for each node, using 17 bytes per node. One byte is used to store the depth level within the octree, 12 bytes to store the coordinates in 3D space, 3 bytes to store the point colour, and 1 byte used as a bit-mask to store information about node children. The data is split into several files (5000 node entries per file, 85kB per file) and files are saved in numerical order, ready for transfer to the remote client.

A WebGL 3D context requests the hierarchical point cloud data via AJAX. Standard HTTP gzip compression is used to compress the files down to around 60% of their initial size. Once the first file is downloaded, points are drawn in 3D space to represent each node of the octree. The points are sized according to the relevant depth level. Meanwhile, further AJAX requests download the rest of the data, one file at a time. As each file is received, the 3D visualisation is carefully updated to display the data. Note that this update process is not simply a case of drawing more data as it arrives, the application must keep track of the highest resolution downloaded so far, and only discard lower-resolution

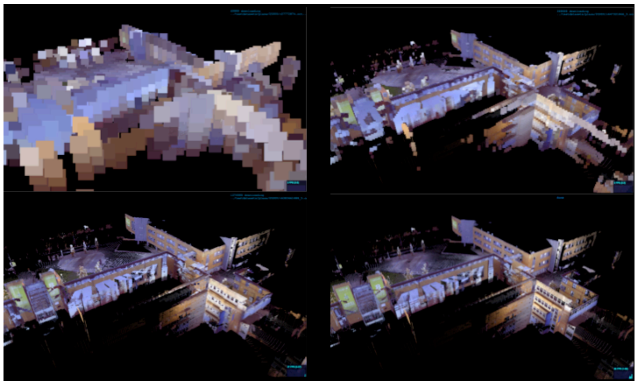

Figure 1: Progressive Visualisation of LIDAR data, with resolution increasing from left to right and top to bottom.

data when it can be sure that no 'holes' will appear in the visualisation.

Figure 1 contains four images demonstrating the progressive nature of the visualisation, with the set gradually being revealed. In terms of speed of visualisation, the initial low-resolution view appears within a second (with a bandwidth of 8Mbps), the more detailed views within a few seconds, and a level of ~3.5 million points (sufficient for good quality representation of a typical LIDAR scene) in about a minute.

2.2. Layered Visualisation of Recorded Footage: matching, registration and timeline interface

Modern digital cinema productions may make use of multi-camera setups, where the cameras are arranged to record the action from several different angles. A registration procedure is required in order to position the cameras in the 3D space, relative to the LIDAR scan, which is taken as 3D ground truth. Structure from motion (SfM) [17] and iterative closest point (ICP) [18] have been widely adopted to register sensor (camera) positions for 2D and 3D modalities, respectively. SfM takes multiple 2D images as input, and matches features to recover camera positions and sparse scene geometry. ICP takes 3D point clouds as input and finds a rigid 3D transformation between two overlapping clouds of points by iteratively minimizing squared-error of registration between the nearest points. Registration of cameras (2D) to LIDAR (3D) is more difficult because their data exist in different domains with different format. We base our application on Kim and Hilton's [19] reconstruction and registration work for multiple modalities into a 3D space. The matching and registration method proposed is hybrid considering both local 3D keypoints and the spatial distribution of neighbouring ones, where the FPFH descriptor is used to extract the relationship of neighbouring keypoints in order to support local feature matching. This leads to more robust matching, effective and efficient in cross-modal registration on LIDAR scans, multiple photos, spherical scans and RGBD video data. The result leads to a transformation matrix for each camera that contains both the position and rotation of each camera (in the case of multiple cameras setups), registered to the LIDAR data.

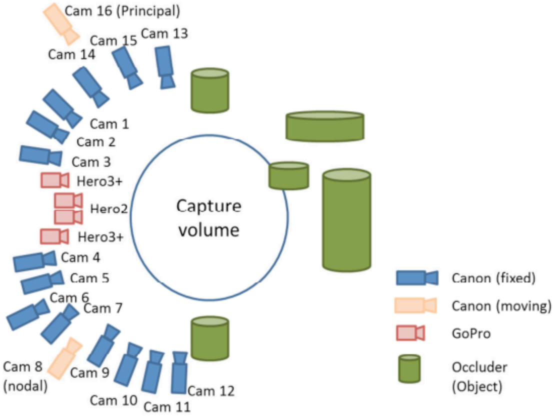

Figure 2: Figurative representation of a typical multicamera setup

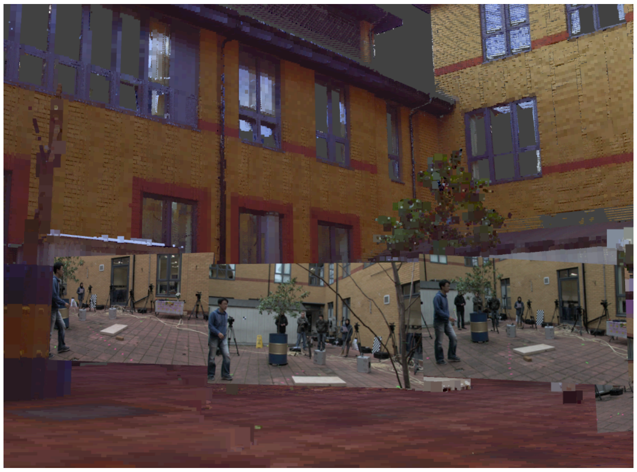

Figure 3: Figurative representation of a typical multicamera setup

To visualise the cameras in our on-set 3D system, we create a simple plane geometry for each of the cameras and position it in the scene using the transformation matrix extracted via the registration process. In order to view the footage recorded by each camera in the 3D scene, we regularly sample the frames of footage and use them as 3D texture information for planes. To ensure the frame-rate is adequate for real-time viewing, a script-based system, using standard tools, compresses the camera footage down to a resolution suitable for web-display, and saves the video using the OGG/Theora codec. For each camera in the scene, a HTML5 video element is created, and at playback each frame is extracted from the video and applied as a texture to the plane representing the relevant camera in the 3D scene (see Figure 3). To control camera selection, playback and

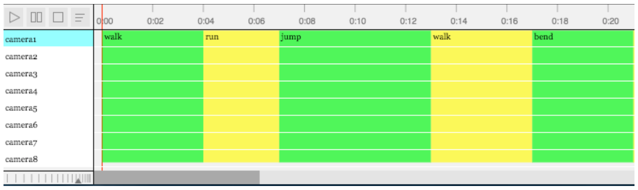

Figure 4: Timeline interface segmented according to primary actor action metadata

scrubbing, a custom timeline interface has been developed in Javascript. The timeline permits the user to select the camera and view it in 3D space, play/pause/stop the action, and scrub to any part of the footage (see figure 4).

3. METADATA VISUALISATION: ACTIONS, SALIENCY, COVERAGE

The actors and human actions in multi-view footage can be recognized through the use of clustering techniques especially accelerated to deal with big media data on set as well as the camera which is most salient with respect to an action, with a performance competitive with the state of the art [20]. Saliency could be used, for instance, to decide which shot should be included when editing in post-production; metadata are useful to solve more intelligent queries. The metadata resulting from the processing, is provided as an XML file and associated description with:

- identification of one or more actors in a scene
- identification of actions carried out by the actors, from a subset as described in [21]
- separation of the footage into timed segments
- saliency of the camera
- any additional metadata which has been added manually

This information is currently visualized on the timeline, as shown in Figure 4. Other metadata related to quality is the estimation of camera coverage, for example extracted using [22]. The result of this analysis is a collection of points in 3D space with associated values representing the number of cameras that can accurately see that point. Figure 7 shows how this sparse point cloud can be overlaid onto the set context, allowing users rapid feedback with regards to the coverage of a given configuration of cameras.

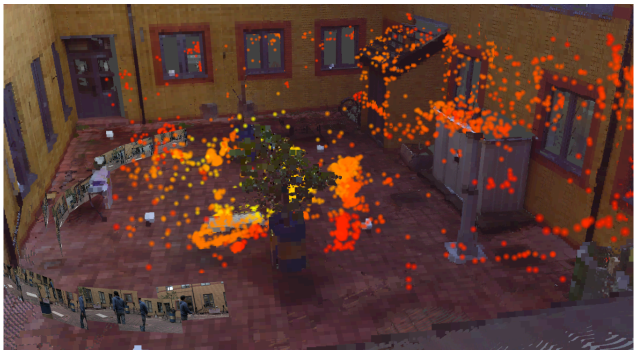

Figure 5: Camera coverage point cloud superimposed on the LIDAR set data. Bright yellow points are seen by more cameras, dark red points are seen by fewer.

4. DISCUSSION AND FUTURE WORK

Typically, academic and professional big data visualizations are largely 2D graphics based, yet professional applications in the media industry are rather more content-based: they use thumbnails, 3D representations of frames+time, and interactive 3D, etc. Our visualization application is consistent with industry practice, and aims at representing the new paradigm of "capturing the whole set", representing data, analyzed data and metadata in a "whole on-set environment". As seen from the examples, it is based (and has been tested) on challenging data that constitute a simulated on-set environment, captured with the support of professionals. The prototype application presented in this paper has been evaluated positively by professional users in the R&D team of Double Negative Visual Effects which has been capturing on-set data for their post-production work in major blockbusters, and has already been included in the development version of the in-house tools used by the company, which the R&D team uses for training the post-production professionals. This seems a strong indication that a professional web-based tool incorporating the features of the prototype would be a valuable and interesting addition to the suite of tools and software that are used, before, during and after production. On the other hand, it shows that the fully working prototype exists and is robust enough to be incorporated into a suite of professional tools, so that further development of a professional level tool is technically feasible. Nevertheless, the prototype in its current form presents several technical and conceptual limitations, which must be considered in future developments. The film industry is accustomed to ever increasing quality and resolution of the data it records and stores; whereas web-based visualization is behind, in terms of quality and speed, of offline solutions (due to both bandwidth and processing capabilities, as discussed previously). In practical terms this means that the tasks where this tool is useful have to be precisely determined, as some tasks will require the highest quality. Related to this issue, users requested to extend the tool with specific functionalities, particularly the ability to view the same data simultaneously on different devices, sharing in real-time annotations and comments between various users. The tool might be thus primarily oriented to provide an integrated and shared "review" of assets, with quality analysis coming both from automated tools and professionals with a bettor integrated tool. This is the prime focus of our future work, which will include intelligent annotation.

ACKNOWLEDGMENTS

Work supported by European FP7 IMPART project (grant 316564), and Spanish EEE (TIN2011-28308-C03-03).

11. REFERENCES

[1] J. Jankowski and S. Decker, “A dual-mode user interface for accessing 3D content on the world wide web,” in *Proceedings of the 21st international conference on World Wide Web - WWW ’12*, 2012, p. 1047.

[2] J. Jankowski and S. Decker, “On the design of a Dual-Mode User Interface for accessing 3D content on the World Wide Web,” *Int. J. Hum. Comput. Stud.*, vol. 71, no. 7, pp. 838–857, 2013.

[3] Unity, “Unity3D.” 2013.

[4] Adobe, “Stage 3D,” 2011. [Online]. Available: http://www.adobe.com/devnet/flashplayer/stage3d.html.

[5] J. Behr, P. Eschler, Y. Jung, and M. Zöllner, “X3DOM: a DOM-based HTML5/X3D integration model,” *Proc. 14th Int. Conf. 3D Web Technol.*, pp. 127–136, 2009.

[6] K. Sons, F. Klein, D. Rubinstein, S. Byelozyorov, and P. Slusallek, “XML3D,” in *Proceedings of the 15th International Conference on Web 3D Technology - Web3D ’10*, 2010, p. 175.

[7] J. Agenjo, A. Evans, and J. Blat, “WebGLStudio – a Pipeline for WebGL Scene Creation,” in *Proceedings 18th International Conference on 3D Web Technology*, 2013, pp. 79–82.

[8] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, and J. Blat, “3D graphics on the web: A survey,” *Comput. Graph.*, vol. 41, pp. 43–61, Feb. 2014.

[9] D. Mongus and B. Žalik, “Efficient method for lossless LIDAR data compression,” *Int. J. Remote Sens.*, vol. 32, no. 9, pp. 2507–2518, May 2011.

[10] B. Merry, P. Marais, and J. Gain, “Compression of Dense and Regular Point Clouds,” *Comput. Graph. Forum*, vol. 25, no. 4, pp. 709–716, Dec. 2006.

[11] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, “Octree-based progressive geometry coding of point clouds,” pp. 103–110, Jul. 2006.

[12] R. Schnabel and R. Klein, “Octree-based Point-Cloud Compression.,” in *SPBG*, 2006, pp. 111–120.

[13] S. Rusinkiewicz and M. Levoy, “QSplat: A multiresolution point rendering system for large meshes,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’00*, 2000, pp. 343–352.

[14] M. Pauly, M. Gross, and L. P. Kobbelt, “Efficient simplification of point-sampled surfaces,” pp. 163–170, Oct. 2002.

[15] A. Evans, J. Agenjo, and J. Blat, “Web-based visualisation of on-set point cloud data,” in *Proceedings of the 11th European Conference on Visual Media Production*, 2014, p. 10.

[16] M. Limper, S. Wagner, C. Stein, Y. Jung, and A. Stork, “Fast delivery of 3D web content: a case study,” in *Proceedings of the 18th International Conference on 3D Web Technology*, 2013, pp. 11–17.

[17] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: exploring photo collections in 3D,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[18] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” in *Robotics-DL tentative*, 1992, pp. 586–606.

[19] H. Kim and A. Hilton, “Evaluation of 3D Feature Descriptors for Multi-modal Data Registration,” in *2013 International Conference on 3D Vision*, 2013, pp. 119–126.

[20] A. Iosifidis, A. Tefas, and I. Pitas, “Minimum Variance Extreme Learning Machine for Human Action Recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[21] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, “Supervised Video Summarization by Content-Based Video Segment Selection,” in *Proceedings of the 11th European Conference on Visual Media Production*, 2014.

[22] E. Imre, J.-Y. Guillemaut, and A. Hilton, “Through-the-Lens Multi-camera Synchronisation and Frame-Drop Detection for 3D Reconstruction,” in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, 2012, pp. 395–402.