

UC San Diego

UC San Diego Previously Published Works

Title

VISUALIZING HIGH RESOLUTION THREE-DIMENSIONAL AND TWO-DIMENSIONAL DATA OF CULTURAL HERITAGE SITES

Permalink

<https://escholarship.org/uc/item/58z4j8k2>

Journal

MEDITERRANEAN ARCHAEOLOGY & ARCHAEOOMETRY, 14(4)

ISSN

1108-9628

Authors

Petrovic, Vid
Vanoni, David J
Richter, Ashley M
[et al.](#)

Publication Date

2014

Peer reviewed



VISUALIZING HIGH RESOLUTION THREE-DIMENSIONAL AND TWO-DIMENSIONAL DATA OF CULTURAL HERITAGE SITES

Vid Petrovic*^{1,2}, David J. Vanoni^{1,2}, Ashley M. Richter^{1,3},
Thomas E. Levy^{1,3} and Falko Kuester^{1,4}

¹ *Center of Interdisciplinary Science for Art, Architecture and Archaeology, Qualcomm Institute, University of California, San Diego*

² *Department of Computer Science and Engineering, University of California, San Diego*

³ *Department of Anthropology, University of California, San Diego*

⁴ *Department of Structural Engineering, University of California, San Diego*

Received: 01/12/2013

Accepted: 02/09/2014

Corresponding author: Vid Petrovic (vipetrov@ucsd.edu)

ABSTRACT

The combination of 3D acquisition (terrestrial and airborne LiDAR, structured light, structure-from-motion) and 2D imaging (photographic, multispectral, panoramic, orthorectified, reflectance transformation) techniques allows the geometry, appearance and other aspects of culturally significant sites to be objectively documented. Traditionally, these data are usually transformed into models such as 3D textured meshes before they are visualized or analyzed—an often time- and effort-intensive process. We propose a system for the direct visualization and analysis of such data, allowing the different aspects recorded to be layered together, and co-visualized with annotations and other relevant information. We describe the required technical foundations, including gigapixel and gigapixel visualization pipelines that enable the dynamic layering of high-resolution imagery over massive minimally-processed LiDAR point clouds that serve as the base spatial layer. In particular, we introduce the pointbuffer—a GPU-resident view-dependent point cache—as the foundation of our gigapixel pipeline, and outline the use of virtual texturing for draping of gigapixel imagery onto point clouds. Finally, we present case studies from sites in Jordan and Italy.

KEYWORDS: visualization; cyber-archaeology; pointbuffer; virtual texturing; point clouds

1. INTRODUCTION

The Roman poet Horace once asked whether given the ability to render a cypress tree, one would opt to include it when commissioned to paint a sailor in the midst of a shipwreck (Horace, 19 BCE). It is a question regarding the over-use of available data: if one has the potential to create visual information—should it be made available or will its additional inclusion be distracting to one's perception of the world around them? For virtual archaeology, there are a multitude of potential data formats that might be dynamically coalesced together to create layers of information, and which, if visualized in a navigable and dynamic format, would significantly aid in appreciating the world around us rather than detract from it. This is especially crucial for cultural heritage explorations where investigative diagnostic imaging results in a multitude of disparate data modalities that are in need of a larger visualization framework in order to experience and appreciate the contextual tapestry of information that might lead to further scientific analysis.

Researchers in cultural heritage have the means of digitally recording various approximate aspects of material reality. Typically this entails first the process of surveying—of applying available digital technologies to acquire the various kinds of objective raw data about the site—followed by a process of systematic survey—of data elaboration and hands-on interpretation that results in the construction of models, maps, reports and other 'deliverables' useful for further study (Bianchini et al., 2012). This process of data elaboration is complex, and is both time- and labor-intensive—accordingly, there have been proposals to formalize the automation of this work (Serna et al., 2012; Pan et al., 2012). However, the challenge persists that much of the digital record remains effectively inaccessible until the products of the elaboration efforts are completed—especially for massive LiDAR scan or Structure from Motion (SfM) imaging campaigns.

Complementary to the business of elaboration and model-making, we suggest that all information, including raw, uninterpreted data, however massive, should be made available for interactive visual review and analysis. This proposed capability can also be useful in the data elaboration process itself. For example, the task of massive dataset coregistration can be made more interactive, allowing the user to verify and refine the alignment among multiple scans, models, and images in real time and at full resolution—and to spot and resolve data quality issues more rapidly.

This paper discusses the on-going development of a layered visualization system that combines disparate data types such as point clouds from terrestrial laser scanning, high resolution photography and multi-spectral imaging in a three-dimensional environment that allows for the co-visualization of annotations and other relevant information (Figure 1). In particular, we outline the gigapoint and gigapixel visualization pipelines that enable the dynamic draping of high resolution imagery over large-scale minimally processed point clouds to create a digital scaffold upon which further information can rest.

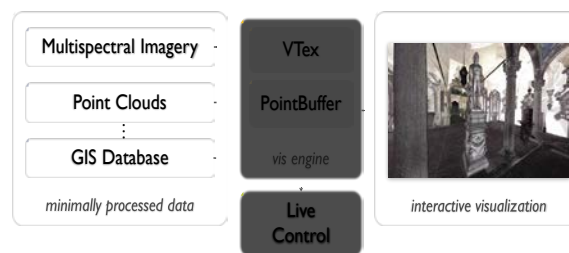


Figure 1 Visualization system overview.

These mechanisms are intended to enable the digital documentation record to be pervasively accessible from the moment of data capture through its stages of elaboration and interpretation, and ultimately academic and public dissemination. To emphasize the utility of this system, case studies of its use on archaeological sites in southern Jordan and historical monuments in northern Italy will be presented.

2. TECHNICAL FOUNDATIONS

The layered system we describe is a collaborative vision between computer scientists, engineers, archaeologists, and historians. Efficient, reproducible, scientific data collection in the field is intended to be channeled into these systems and just as effectively visualized and disseminated. The following outlines the system by addressing the base layer in its data stratigraphy—point clouds—that provide the spatial and geometric foundation for other site data.

2.1 Gigapoint visualization with the *pointbuffer*

Visualization of massive point clouds that are too large to load directly is a well-studied problem (Wand et al., 2008; Scheiblauer et al., 2009; Scheiblauer et al., 2011; Pintus et al., 2011): a common thread is the preprocessing and reorganization of point data according to an out-of-core spatial-subdivision datastructure (commonly octree) to accelerate the selection and loading of the points relevant for visualization. To meet the needs of our project, however, we developed an alternate rapid visualization technique, based on the *pointbuffer* described below, that minimizes data reorganization and preprocessing requirements (relative to existing work) and gives extensive flexibility for fusing different types of data into a single visual representation.

Our technique embodies a shift in strategy away from attempting to transform the input data into the form most optimal for rendering—and focusing more on ensuring adequate rendering performance even with minimally preprocessed data. It leverages the capabilities of modern graphics hardware to flexibly produce high-quality renderings that refine progressively over the course of interaction as data are streamed in from secondary storage. In contrast to most other massive-point-cloud renderers, our system requires only minimal preprocessing of data before visualization—taking time comparable to copying the dataset.

The core of our gigapoint visualization pipeline is the GPU-based *pointbuffer*, a point-caching construct that decouples the interactive performance of visualization from the costs of the streaming large numbers of points from disk or a network server (Figure 2).

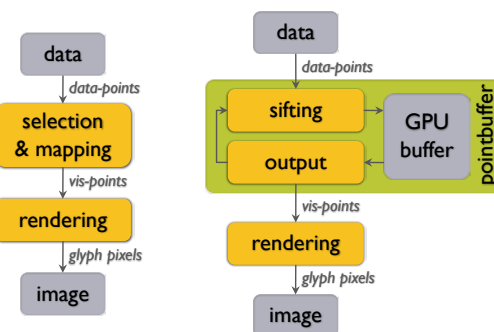


Figure 2 Pointbuffer overview: the pointbuffer accumulates points for visualization, mapping input data to visualization points stored in a view-dependent cache

The pointbuffer explicitly maintains a working set of points needed to render the view, and recycles the points that remain relevant from frame to frame—allowing greater flexibility for combining point and image data, making it possible for imagery to affect both the texture and the underlying geometry being visualized.

Note that caching is a commonly applied performance optimization: a particularly fine-grained example is employed for point rendering by *deferred splatting* (Guennebaud et al., 2004), a generalization of deferred rendering that uses pixel-binning of point indices (references to points stored in memory buffers) to perform high-precision final point selection, and further reduces the splatting workload by re-projecting and reusing indices from the previous frame in performing visibility-splatting for the current frame. The pointbuffer we present here simplifies, generalizes, and extends this deferred splatting technique, making it possible to use point-caching as the primary mechanism for building a gigapoint visualization pipeline.

We demonstrate that for large real-world datasets, effective exploitation of temporal coherence allows minimally-reorganized

point collections to be interactively explored, with immediate control over how the input data are mapped to visual elements. We describe some details of the pointbuffer in Section 3.

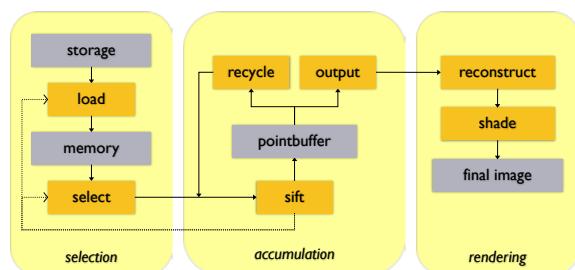


Figure 3 Pointbuffer processing stages.

2.2 Gigapixel texturing

To enable the use of massive imagery in visualization, we employ a virtual texturing technique (Lefebvre et al., 2004; Taibo et al., 2009; Mayer et al., 2011), a combination of classical MIP-mapping and virtual memory approaches. The approach is to store images in a tiled multi-resolution pyramid, and load only the tiles needed to perform the texture lookups requested. For a given view, the set of tiles corresponding to the observed texture lookups (2D+level-of-detail/scale coordinates) is determined. In our variant of virtual texturing, this step is performed entirely on the GPU, with only the results being read back by the CPU – permitting a tight coupling with the gigapixel pipeline. The needed tiles are then fetched from disk and uploaded to the GPU tile cache. We keep the peak of the image pyramid always resident so that all lookups can be satisfied (albeit at a lower resolution) even before the more appropriate are uploaded, and perform trilinear filtering between the two bracketing levels-of-detail once the corresponding tiles are cached.

3. THE POINTBUFFER

The *pointbuffer* is a GPU-based datastructure that decouples the loading and rendering pipeline stages by buffering the visualization points needed for rendering. Vispoints remain in the pointbuffer for as long as they are appropriate for the view

being rendered. Whenever the view setup changes—due to camera motion, or a change in selection, visualization or mapping parameters—the pointbuffer keeps the vispoints that are still appropriate for the new view.

The *pointbuffer* offers two high-level operations: sift and output. The sift operation processes a datapoint batch, adding to the buffer the visualization points appropriate for the view, and reporting as feedback the number of points added. The output operation streams out, in part or in full, the buffered vispoints for further processing or rendering. The points in the buffer are recycled between frames, accounting for any change in view. This can be performed by ping-ponging between two buffers: points are output from the old buffer and sifted into the new buffer.

A sifting cycle consists of two steps:

1. Recycle points already in the buffer.
2. Sift in batches of new points.

Points accumulate in the pointbuffer over many frames, allowing the resulting renderings to refine gradually over time, even

3.1 Pointbuffer binning

Points in the pointbuffer are stored in a grid of bins, with at most one point per bin, with the binning scheme determining the mapping of image point samples to image pixels. In the simplest scheme, the binning grid coincides with the image pixel grid, allowing the pointbuffer to hold at most one vispoint per image pixel. More elaborate binning schemes are preferable in practice, with multiple bins per pixel, or a nonuniform distribution of bins, such that there are more samples per pixel near the center of the image than elsewhere, for example. The binning scheme and capacity—the number of bins—can be set independently of the output image size, and is a parameter that influences performance and attainable rendering quality. In practice, image-center-biased bin distributions provide the best quality-performance bal-

ance, with oversampling near the image center (for crisp, low-alias detail rendition) and a manageable total bin count (for performance).

3.2 Point selection and loading

For this work, we understand a point dataset to be a collection of distinct point sets, or point clusters. We ensure through preprocessing (described below) that each cluster is stored as a sequence of points such that:

1. any contiguous subsequence of a cluster is expected to have approximately the same spatial distribution; and
2. different clusters ideally have different spatial distributions.

The first property above allows us to use a contiguous subsequence of cluster points, such as a prefix sequence, to render an approximation for the cluster. It also makes it possible for us to estimate the contribution that points from a cluster will make to a final rendered image by measuring the contribution made to an approximate image by points from a smaller prefix sequence. The second property differentiates the clusters from a selection standpoint, allowing us to make gains in loading efficiency by view-dependently favoring selection from some clusters over others.

The contribution that the points from a single cluster make to the rendered image is view-dependent. We quantify this cluster contribution as the number of points binned from the cluster. For any given view, clusters in general have differing levels of contribution. For close-up views, points from relatively few clusters tend to dominate, with most clusters contributing few or no points. For total views, in which most of the scene is visible, it can happen that many or all clusters contribute relatively few points each.

The selection engine focuses the loading and sifting effort on the clusters that contribute the most to the image. Initially, the clusters are assumed to make contributions proportional to their point counts. As

points are sifted into the pointbuffer, the feedback returned—the number of points accepted into the pointbuffer—is used to estimate each cluster’s contribution, and to guide the selection process over the following sifting cycles.

3.3 Dataset preprocessing

The purpose of preprocessing is to generate and store the datapoint clusters in a form suitable for selection. Many kinds of point datasets are already stored, or can be efficiently exported, as sequences of datapoints exhibiting at least some spatial coherence. Examples include LiDAR data (stored in scan order), volume data (stored in slices or blocks), image data (e.g., SfM), and other spatially-organized datasets. Transcoding such a dataset is particularly simple and fast, since the data can be broken up into clusters sequentially. The order of points in each cluster is then randomized, yielding a point cluster with the desired characteristics.

The preprocessing procedure is:

1. Fill an array of the desired (cluster) size sequentially with datapoints, optionally transforming or reformatting the points.
2. Shuffle the points in the array.
3. Store the array as a cluster.
4. Repeat steps 1-3 until all points have been processed.

Note that the transcoding time is linear in the number of points. Since the points in each cluster are shuffled, any prefix sequence for the cluster is a random subset of the full point set, and will therefore on average have approximately the same spatial distribution as the full set.

4. PERFORMANCE CHARACTERISTICS

On a midrange laptop (ca. 2011, Intel Core i5-254M, 4GiB RAM, 2.6GHz, AMD Radeon HD 6750M with 512MiB graphics RAM), the gigapoint pipeline maintains rates above 30Hz at 1280x720 (for all datasets tested, with largest at ~2.75 billion points), and the virtual texturing pipeline

maintains rates above 60Hz at 1920x1080 (tested with synthetic data for ~16 billion pixels). The combined pipelines have performance similar to that of the point pipeline alone. Renderings usually refine over the course of a few seconds at most for point rendering, and well under a second for texturing.

4.1 Preprocessing performance

Preprocessing time is dominated by the cost of reading the raw data and writing the output clusters: the point reformatting and shuffling costs are generally dwarfed by I/O costs. The preprocessing rate depends upon the input data format, the size of the output point format, and the speed of the source drive (and destination drive, if different). The rate ranges from 0.5 million points per second for transcoding text PTX files using a single external USB drive to several million points per second with high-performance drives. The overhead relative to simply reading from the source and writing to the destination is in our implementation caused by input parsing inefficiencies and insufficient I/O overlapping. We expect that further implementation optimizations can bring the transcoding costs down to near that of a direct copy operation.

4.2 Selection performance

The loading of cluster blocks is in practice done at the disk's maximum read rate, or at a user-specified lower rate. The amount of time it takes for a given view to fully refine depends on how different the view is from recent views and the level of main-memory caching that has been attained. As the rendering refines, the progress that is being made is visually evident, and this fact tends to make subjectively acceptable even longer waits than are encountered in practice.

The refinement time is the greatest after a cold start, when all data must be loaded from disk. The system tends to reach steady-state after less than a minute of interaction (usually in under 30 seconds);

overall, refinement rarely takes more than a few seconds.

5. CASE STUDIES

5.1 The Byzantine Church at Petra

The current test-case for our system is to combine terrestrial laser scanning point clouds with systematic high resolution photography, diagnostic maps, and corresponding semantic information collected during and for the 2012 diagnostic imaging survey of Petra's Byzantine Mosaic Church (see Figures 4 and 5) at the behest of the American Center of Oriental Research's Temple of the Winged Lion and Environs Conservation Projects (Levy et al., 2013). In this case, our work is aimed at creating not just a digital tourist tool for the Petra UNESCO World Cultural Heritage Site in Jordan, but in building a visualization tool that can be useful for the on-going research and conservation monitoring at the site.



Figure 4 The Byzantine Church at Petra, Jordan: LiDAR of mosaic.



Figure 5 The Byzantine Church at Petra: LiDAR of mosaic with photographic overlay.

Acquisition of data for the Byzantine Mosaic Church was a field trial of rescue

archaeology methodologies under development, and was performed in under two hours. Four researchers (one LiDAR operator, two photographers, and one assistant) worked concurrently to maximize coverage of the site within the window of time available. Over 3000 digital photographs were taken in a pattern suitable for SfM camera pose estimation and 3D modeling.

The images were subsequently processed using Agisoft PhotoScan (a commercial SfM/photogrammetry software package) to produce high-resolution (32kx32k pixel) orthophotos of the floor mosaics, which were then draped dynamically onto LiDAR data using the system described in this paper. Note that it is also possible to drape images individually onto the point cloud using the camera parameters (intrinsic and pose) estimated by PhotoScene, bypassing the generation of orthophotos.

5.2 Salone dei Cinquecento, Palazzo Vecchio

Our second example is the diagnostic imaging study of the Hall of Five-hundred at Palazzo Vecchio, in Florence, Italy. This Renaissance site is the possible location of a Leonardo Da Vinci painting (the Battle of Anghiari) whose whereabouts have been unknown for over 450 years. Over one billion LiDAR points were collected, with 1mm-resolution capturing the six frescoes and eight sculptures along the east and west walls. Additionally, high-resolution (~30kx20k pixel) visible-light imagery was captured by a Panoscan rotating line-scan camera, as well as a thermal image mosa-

ics. In Figure 6, the LiDAR data is shown layered with the Panoscan imager and thermal annotations (in green). A preliminary version of the system outlined in this paper (with additional layering of CAD models of the Palazzo and ground-penetrating-radar imagery of the wall and Vasari fresco) was used to help guide an endoscopic study in 2011.



Figure 6 Battaglia di Anghiari project: high-resolution visible-light image captured by a Panoscan rotating line-scan camera, and a thermal image mosaic (as a green overlay), on top of a LiDAR point cloud.

6. CONCLUSIONS

We have described a small set of mechanisms sufficient for dynamically visualizing minimally processed 3D point clouds along with high-resolution 2D imagery. The approaches presented—pointbuffer-based gigapoint and virtual-texturing-based gigapixel pipelines—allow massive digital documentation datasets, comprising LiDAR and multispectral photography, to be viewed and inspected interactively throughout their digital lives, from acquisition to elaboration to dissemination.

REFERENCES

- Bianchini, C., Borgogni, F., Ippolito, A., Senatore, L.J., Capiato, E., Capocéfalo, C. and Cosentino, F. (2012) From surveying to representation theoretical background, practical issues, possible guidelines. *Virtual Systems and Multimedia* 2012.
- Guennebaud, G., Barthe, L. and Paulin, M. (2014) Deferred splatting. *Computer Graphics Forum*, 23(3):653–660, Sept. 2004.
- Horace, *Ars Poetica* 21.
- Lefebvre, S., Darbon, J. and Neyret, F. (2004) Unified texture management for arbitrary meshes. *Research Report 5210*, INRIA.

- Levy, T.E., Tuttle, C.A., Vincent, M., Howland, M., Richter, A., Petrovic, V. and Vanoni, D. (2013) The 2012 Petra Cyber-Archaeology Cultural Conservation Expedition: Temple of the Winged Lions and Environs, Jordan. *Antiquity (Project Gallery)*, <http://antiquity.ac.uk/projgall/levy335>.
- Mayer, I., Scheiblaue, C. and Mayer, A.J. (2011) Virtual texturing in the documentation of cultural heritage the Domitilla catacomb in Rome, *Proc. Of XXIIIrd International CIPA Symposium*.
- Pan, X., Schiffer, T., Schrottner, M., Havemann, S., Hecher, M., Berndt, R. and Fellner, D.W. (2012) A scalable repository infrastructure for CH digital object management. *Virtual Systems and Multimedia 2012*.
- Pintus, R., Gobbetti, E. and Agus, M. (2011) Real-time rendering of massive unstructured raw point clouds using screen-space operators. *The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*.
- Scheiblaue, C., Zimmermann, N. and Wimmer, M. (2009) Interactive Domitilla Catacomb exploration. *The 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*.
- Scheiblaue, C. and Wimmer, M. (2011) Out-of-core selection and editing of huge point clouds. *Computers & Graphics*, 35(2):342-351.
- Serna, S.P., Schmedt, H., Ritz, M. and Stork, A. (2012) Interactive semantic enrichment of 3D cultural heritage collections. *The 13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*.
- Taibo, J., Seoane, A. and Hernández, L.A. (2009) Dynamic virtual textures. *Journal of WSCG*, Volume 17, Number 1.
- Wand, M., Berner, A., Bokeloh, M., Jenke, P., Fleck, A., Hoffmann, M., Maier, B., Staneker, D., Schilling, A. and Seidel, H. Processing and interactive editing of huge point clouds from 3D scanners. *Computers & Graphics*, 32(2):204-220.