The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2004)
K. Cain, Y. Chrysanthou, F. Niccolucci, N. Silberman (Editors)

A Point-Based Approach for Capture, Display and Illustration of Very Complex Archeological Artefacts

Florent Duguet[1,2] † and George Drettakis[1] and Daniel Girardeau-Montaut[2,4] and Jean-Luc Martinez[3] and Francis Schmitt[2]

[1] REVES/INRIA Sophia-Antipolis, [2] ENST Paris, [3] Ecole Française d'Athènes, Musée du Louvre, [4] EDF R&D

Abstract

In this paper we present a complete point-based pipeline for the capture, display and illustration of very large scans of archeological artifacts. This approach was developed as part of a project involving archeologists and computer graphics researchers, working on the Delphi "Dancers Column". We first determined the archeologists' requirements for interactive viewing and documentary illustration. To satisfy these needs we use a compact point-based structure of the very large data, permitting interactive viewing in 3D. This helps the archeologists to examine and position the fragments. We introduce efficient construction algorithms for this structure, allowing it to be built on limited-memory platforms, such as those available on the field. We also propose a new stylized rendering approach based on an inverse cylindrical projection and 2D skydome rendering. This illustrative style has been used as a planning tool for fragment docking and as a substitute for traditional illustration in an archeological publication. Other uses of these tools are currently under way in the context of this project.

1. Introduction

The work described here is a collaborative project involving archeologists and computer-graphics researchers. The project involves a specific study concerning the Ancient Greek "Dancers column" monument in Delphi (see Fig. 1). At the outset, the archeologists required a detailed 3D scan of the monument, the ability to interactively view the result and a way to obtain stylized illustrations. Based on these discussions we concentrated on three main tasks in terms of archeological research. The first was providing a tool which would permit better understanding of monument reconstruction hypotheses in 3D. This required an interactive viewer with the ability to handle the very large datasets involved. The second was the preparatory study of fitting for different fragments; our solution was the generation of appropriate non-photorealistic illustrations, in a style which helped in overall understanding. The final requirement was to find an alternative for traditional hand-drawn illustrations of such complex structures. This need was particularly important since such illustrations will be used as part of an appropriate archeological publication (book).

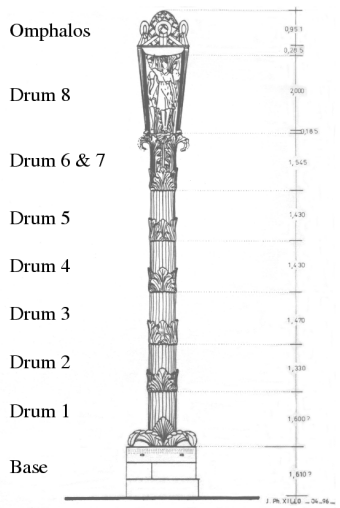

Figure 1: *A recent attempt at the illustration of the Dancers column (drawing by J.-Ph. XILLO, 1996).*

The main technical challenge in this work was the difficulty implied by the size of the acquired data sets, and the

† Florent.Duguet@sophia.inria.fr

implications this had on the choice of storage and display algorithms which can be applied.

The archaeologists in this project were somewhat concerned with the resolution of the 3D data. If the resolution of the scanned data is too coarse, the meshed version of a sparse set of points results in a faceted look whose edges prevent the archeologists from perceiving the reality they are used to studying. They often prefer the display of the raw data (the 3D points) without any meshing when they are dense enough, or when they are rendered with appropriate styles. The Digital Michelangelo Project opened a new research direction in this field [LPC*00]. We have followed their pioneering philosophy, and extended this approach as a result of our close multi-disciplinary collaboration. The archeologists' requirements, the fact that the original scanned data is in the form of points, and recent advances in point-based methods both in terms of storage and rendering, led us to propose a point-based representation throughout the pipeline.

We will show how to organise the scanned 3D points efficiently in appropriate packed 3D spatial subdivision structures. Two approaches are presented for the construction of this structure, taking into account the size of the data processed. We then present an application of this data structure, for interactive display. In addition we propose a new stylized projected 2D rendering of very large data sets, based on skydome rendering. We will refer to these renderings (for examples see Fig. 9,11,10) as *développées* in what follows.

Computer graphics researchers and archeologists have worked hand-in-hand throughout this project, allowing us to develop computer graphics (CG) solutions adapted to the needs of the archeologists. In the results section we show how the interactive viewer and illustrations have already helped in the process of archeological research in our "Dancers Column" project. Further use of these results is planned in the near future.

1.1. The "Dancers Column" Project

The dancers column from Delphi in Greece (Fig. 1), 14 meters high, is a major artifact of Greek art since it is recognized as the first acanthus column ever sculptured. Seven drums plus that of the dancers and the famous "Omphalos" completed by 3 large leaves at the base level, form this tall column. More than 200 fragments are known. The dancers column has not yet been published as a book because of its geometrical complexity: the size, weight and fragility of each drum and fragment prevent any human from handling them to manually re-erect the monument. Besides, the unanswered questions are tough: what are the correct relative positions of the 8 drums and the Omphalos to be coherent with the supposed ternary rhythm from drum to drum with growing size of acanthus leaves at each level of the column?

One important objective of our project is thus to publish the state of its documentation and to dispute the proposed hypothesis. A 3D archive could also be useful since Delphi is located in a seismic area.

The archeologists need details with millimetric precision, especially on the broken faces of fragments and for the grooves since these are leading cues for the docking of the various fragments. They have been carved with a millimetric pattern (see Fig. 2). The distance between 3D points was chosen smaller or equal to 0.5 mm. This gives 300 million 3D points. The dispersion error was chosen smaller than 1 mm (standard deviation smaller than 0.33 mm) and the quality of registration of the 3D frames was specified to be smaller than 1 mm. A lower resolution, a larger dispersion or an inferior registration quality could notably blur the art of the ancient artists.

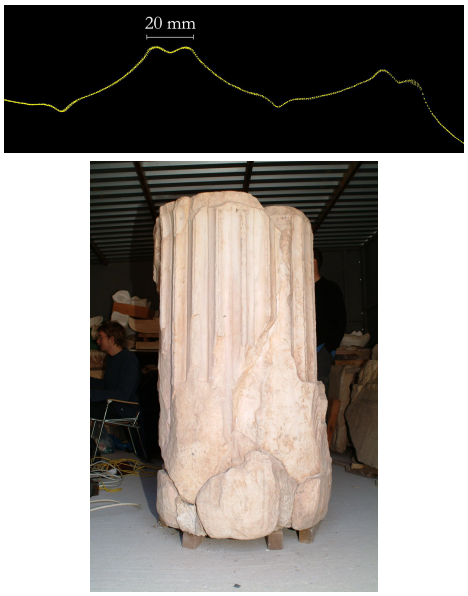

Figure 2: *Top: a graph indicating the size of the braced grooves. Bottom: Drum 2 with its grooves (see also Fig. 9).*

1.2. User needs of the Archeologists

The final documentation of the monument will take the form of a book [Mart04]. In this kind of publication, the archeologists have to draw the components of the column with their fragments in place. One way this is done is to project the cylindrical shapes of each drum and the ovoid shape of the Omphalos onto a plane, in order to get the 2D projections for the *développée*.

The repositioning of the fragments also requires the projected illustrations plus interactive viewing and docking: the archeologist first searches in the *développée*, for the possible positions of the current fragment and then verifies the fit using the 3D archive.

In what follows we will show how to produce alternatives to these manual techniques. It is important to note that all technical choices were made jointly by CG researchers and archeologists.

2. Previous Work

In terms of computer graphics research, this work has been inspired by previous research in point-based geometry and skydome/terrain rendering.

2.1. Points

In the last five years, Point-Based Geometry has become a very popular 3D object representation for geometry processing and graphics [AGP*04]. The design of processing and rendering tools using this geometry representation is relatively straightforward, making simple rendering systems easy to implement. Even though graphics hardware rendering pipelines have been designed for polygons, the rendering of points is even easier than for polygons.

Points have been used for rendering in various contexts, and in various forms. A point can be an impostor for a small piece of geometry [RL00], a sample of input data (scanning), or some sample of the geometry generated on the fly procedurally [SD01]. It might carry several types of data, such as the surface normal, color, transparency, for the surface sample itself. Additional data may also be stored if it is an impostor for a larger piece of geometry.

While this representation is simple, it requires a large number of samples to be accurate for complex objects. Consequently this quickly becomes the drawback of the approach. Several papers focused on packing and structuring the points such that the representation storage overhead is minimal. In particular, there have been different contributions to develop hierarchies of points, making the point storage more efficient and multiresolution [RL00, BWK02, DD04].

2.2. Skydome and Terrain Rendering

The stylistic rendering we will be introducing is related to skydome rendering. There is a large body of literature in this domain, and it is beyond the scope of this paper to review this in detail. Horizon maps were introduced by Max [Max88], who approximated the computation of shadows on a bump map by precomputing a sampling of 8 directions. Much work in optimization of this approach has followed, notably work by Stewart [Ste98] which uses hierarchical analytic computation to compute the self shadowing of a terrain, and that by Sloan and Cohen [SC00] which uses graphics hardware to accelerate the computation of bump map shadows. A graphics-hardware accelerated technique for skydome rendering in archeological applications was presented in [BCS01].

3. Overview of the entire pipeline

Our pipeline starts with the capture phase, resulting in aligned scans of 3D points, which are then encoded efficiently into a compact point-based structure. This structure can then be used for interactive 3D rendering. We also use this data to generate 2D illustrations using an inverse cylindrical projection and skydome rendering. An additional 3D skydome rendering mode is also presented. Finally, the archeologists use the resulting images and tools to help them in their work.

3.1. Capture

A team of 8 people traveled to Delphi with 2 MINOLTA VI910 scanners. One Minolta VI910 was dedicated to the Dancers in order to avoid cross calibration difference between scanners. The conditions of intervention were particularly difficult, since the Museum of Delphi was in complete recasting for the Olympic Games (Fig. 3): dust and power failure were the daily enemies. For the rig required to scan Drum 8 (the Dancers), a simple, cheap and stable design was proposed by Insight†, which used inertial damping to come to rest in a relatively short time. To evaluate "noise motion", several point lasers were mounted near the scan head and aimed at walls located 10m away in line of sight. In this way, even subtle motion of the scanning rig would cause pronounced motion in the laser dots. The ground crew used these lasers to determine if the rig was stable before each scan. The Scanning, Data and Registration officers formed a sub-team dedicated to each scanner (Fig. 4). The Scanning officer was responsible for the correct positioning and stability of the rig and for the orientation of the scanner. The Data officer monitored the acquisition of data and had to rapidly check if the data were correct in terms of dispersion. The Registration officer registered the incoming frames on site, in order to verify the distortion error. If the frames were too oblique with respect to the surfaces scanned, or if the spatial extension of the scanned area was too large, the distortion in the field of measure increased rapidly. In this case, a revised scan was requested.

3.2. Data Structures and Interactive Rendering

We use the data structure of [DD04] for our point-based representation of the data since it allows flexible multi-level rendering with small overhead. We used an octree for reasons given later in the paper. We adapted the algorithm of [DD04] for our purposes since it could only process surfaces and we have points as input.

Given the input data, we insert points and normals into an octree structure. The point position leads to a leaf cell in the octree and the normal is encoded by recursive triangular

† http://www.insightdigital.org/

Figure 3: *Capture was done in relatively harsh conditions.*

Figure 4: *The capture team at work on the site.*

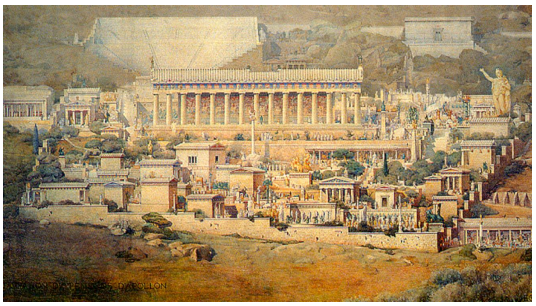

Figure 5: *A 19th century drawing of Delphi site.*

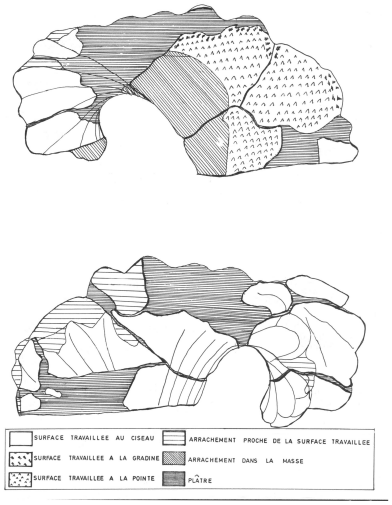

Figure 6: *Attempt at manual drawing of a acanthus leaf.*

subdivision of the octahedron using 15 bits. 15 bits provided good visual results. Each leaf cell might receive more than a sample point, in which case we weight-average the current cell normal with the new normal (weights are the number of normal inserted). Once the leaf level is built, we pull the normals from the leaf cells up to the root cell by performing a weighted average of the child cells (from 1 to 8 child cells). We finally obtain the representation presented in [DD04].

This structure can then be used for flexible interactive rendering as described in [DD04]. By choosing an appropriate level of the structure we can achieve good frame rates.

3.3. 2D and 3D Skydome Rendering

The choice of the skydome algorithm was dictated by the archeological needs: the archeologists appreciate the artistic rendering dating back to the 19th century (Fig. 5), because of its expressive power due to the high degree of detail and the quality of the shadows [RD03], even if they know that this is an interpretation of the real artifacts.

The skydome algorithm applied on a dense cloud of points has the ability to provide a 19th century look without the subjective filtering of the draughtsman. As the drums and the leaves have been restored during the first half of the 20th century, and these restorations are partially damaged, the archeologists have to draw which parts of the elements are made of preserved or fractured marble, and which ones are made of plaster (Fig. 6). Besides, surfaces have been carved using three different tools: point, flat chisel and claw chisel (indented). The latter was used in order to enhance the fixing of the drum ones on the others (the same for the fixing of the 3 big leaves on Drum 1). The drawing has to separate these different types of materials and carving techniques. This explains why stippling techniques are not well suited for the drawing of these various types of data. On the contrary, the skydome algorithm provides us the very fine rendering of the data and allows us to perceive plaster versus marble, preserved or fractured, regular, picked or indented.

To produce such images, we project the points on a cylinder, and store them in a height field. Visually, we do as follows: we cut the cylinder vertically on a side, and we get the resulting four corners on a rectangle with minimal stretching of geometry. The result is a 2D height function, sampled by points (see Fig. 7). We thus obtain an image of heights. We can visualize this height field in several ways: height, di-

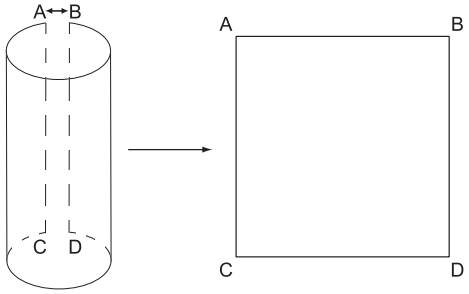

Figure 7: *Unfolding a cylinder to a plane.*

rectional shading, skydome shading. The first considers the height as the luminance of a gray-scaled image, the second shades the plane with a single directional light source, and the last one lights the plane with a uniform white skydome lighting (cloudy day). Of these solutions, skydome rendering was chosen by the archeologists since it provided the best visual quality, and understanding of local geometry.

It is worth noting that *développées* are not real terrains since their geometry is warped due to the non linear projection, and it thus makes no sense to focus on their exact rendering. The approximate skydome rendering provided good results, the local aspects of the object being well preserved on regions of interest, while the global structure remains well understood.

3.4. Use by the archeologists

The archeologists in our project used both the interactive viewer, which allows a better understanding of spatial relationships and the 3D natures of the scans, and the skydome rendering both as an alternative to illustration and a support-mechanism for perceiving and understanding the spatial relations between the fragments.

4. Packed point representation

The main difficulty in the approach is the amount of data. For individual drums, the number of points easily reaches 30 million, and the data precision is in the order of $1/10000^{th}$ of the size of the object. We thus wanted to work with octrees of level 13 ($2^{13} = 8192$) to minimize the additional error incurred by the snapping of points to the center of a leaf cell. Simple calculation shows that for a surface, about 4 child per cell are full, leading to approximately $\frac{4^{14}-1}{3} = 89,478,485$ octree cells ([DD04]). If we allow 1Gb of memory (which is now easily available on laptops), this leaves us with at most 11 bytes per cells. We thus need to pack our data smartly.

The structure of our cell is as follows:

- 8 bits: childhood code
- 16 bits: normal
- 32 bits: a pointer to children cells (for non leaf nodes), or a weight (for leaf nodes)

This leads to 7 bytes per cell. Given this size requirement, we need to adapt standard system memory management since its direct use would be particularly inefficient. Recall that the data-sets being treated are very large, and thus that any overhead results in significant wasted memory; this can make the difference between the data fitting in memory or not.

The way we proceed is to add an additional level of memory management, which allocates memory in large chunks, and performs efficient memory handling.

We developed two algorithms: one is optimal in terms of size (Optimal algorithm), and the other can operate directly on point streams (Streaming algorithm), requiring only a single traversal of the input data.

The childhood code contains eight bits which describes the occupancy (1) or vacancy (0) of the child cells. It has been already used in this context in [BWK02]. The bounding box of each cell is not stored since it can be computed on the fly during traversal. Only the bounding box of the root cell is required.

4.1. Optimal Algorithm

The Optimal algorithm is iterative, with one pass per level of the final octree. We initialize it with the root cell and construct it level by level by subdividing the parent cells of a given level l into their children cells of the next level $l+1$, up to a given level L where all cells are leaf cells without children.

For each given existing level, we traverse the entire point set. For each input point, we find the corresponding leaf cell, and set the childhood bit of the sub-cell which should be created for this point in the next level down. We call this the marking pass.

Then, the octree is traversed to allocate tables of children, starting at the root node. Given the preceeding marking operation for this level, we know the size of the tables of children required for each leaf cell. At each leaf cell, we use our memory management layer to allocate the exact amount of memory required for the table of children.

Our memory management allocates cells by big chunks, and returns a pointer to a part of the chunk, which can be read as a table of cells. This leads to a very small overhead if the chunks are big-enough (2k cells is typical). The overhead is the system size for table allocation plus handling of chunks (12 bytes per chunk) plus the lost end of a chunk (maximum of 7 cells per chunk). This approach thus provides near optimal memory allocation.

Once all allocations have been done, the points are inserted in a last pass, by traversing the octree, and finding the appropriate leaf node corresponding to the point being inserted.

4.2. Streaming Algorithm

The second algorithm is not optimal in memory allocation, and is a bit harder to implement. However, if the data is well arranged, it can achieve good results. The algorithm is in one pass, with a single function: Populate. A cell is populated up to a given level. If the child is not allocated, the cell is modified such that the child is allocated. The Populate routine is as follows:

```
Populate(point, normal, level)
if intermediate level // level is not zero
   Identify sub-cell
   if sub-cell exists
      sub-cell->Populate (point, nor-
mal, level-1)
   else
      Decode child table to temp structure
      Free child table
      Update temp structure with new child
      Reallocate bigger child table
      Encode child table
      sub-cell->Populate (point, nor-
mal, level-1)
   end
else // we are at leaf level
   InsertLeaf (point, normal)
end
```

The main difference between the two algorithms is the need to decode and encode on the fly and the reallocations implied by these operations.

It might appear that the Streaming algorithm is faster than the Optimal algorithm. Most often, this is not true due to the Free/Reallocate processing which has to smartly handle the release and allocation of small chunks of memory of different size. It might also be a source of large memory allocations, which could not be freed. For example, random insertion is a worst case configuration since many child-cell tables of size 1 would be allocated, but never actually freed from the tables.

4.3. Pull Normals

The Pull algorithm is common to the two construction schemes since it does not do any allocation. A weighted average of the the normals of the sub-cells is computed for each cell. The resulting normal is then renormalized, and the cumulative weight of its children is pulled up from the sub-tree; this is a post-processing step, attributing normals to non-leaf nodes without storing weight.

4.4. Storage

For the storage, we do not keep track of the weight, and thus discard this data from the structure, allowing a big save in the memory consumption The output structure is thus much smaller (approximately a factor of 3) than the online construction structure. When reading the object, the weight is not present, and we thus generate different leaf cells, which do not have the same structure size, allowing a much lighter representation.

4.5. Interactive 3D Rendering

For rendering, we use an approach similar to [DD04], but since the target platform is a desktop or laptop PC, we instead take advantage of the graphics card. The octree is traversed and points are generated for each cell at the user-defined level: the coordinates are given by the center of the box of the cell (computed on the fly), and the normal is decoded as well. The user-defined level is coarser than the level of the octree, and splats are rendered as impostors for subtrees (which are under the user-defined level). The points and normals are stored in Vertex and Normal Arrays (OpenGL structures), and rendered using graphics hardware. The primitive used is *GL_POINTS*, and a splatting radius can be defined, for an impression of dense sampling, even though it might result in a more blocky appearance. Using a common level (8 to 10) for the real-time display of the model, we can achieve very good frame rates (60 to 10 frame per second respectively on a Pentium IV with a GeForce 4 card).

We only render impostors of the real data. We can switch to a more accurate rendering using all the available data, at the cost of interactivity. For this approach, instead of generating a single array for the whole data (which would not be possible for very big models), we generate smaller arrays and render those as they are generated. We can finally render tens of millions of points incore, using graphics acceleration. This rendering is not interactive, but is used to generate high quality images (at high resolutions).

5. Skydome Algorithm

Shading an object with a light source, as those available by using OpenGL local shading models (Point/Direction/Spot lights), can generate high quality images, but some details and features of the object are missing. Raytracing would be possible, but for complex light models, such as ambient occlusion rendering, this solution is very expensive. To our knowledge, an efficient solution for ray tracing tens of millions of points has not been previously presented. Although this should be possible, using for example Moving Leasts Squares [ABCO*03], implementation issues arising from the huge amount of data could make these algorithms very slow. Also, the resulting surface would be an approximation of the point set with their normals (which are better estimated using individual scan shots rather than using the registered multi-scan point data).

We have thus adopted an image-space illumination algorithm to tackle the complexity problem. This image-space skydome algorithm will be used for illustration, resulting in renderings which satisfy the needs of the archeologists.

5.1. Construction

For these computations, we work in image space, where the image is a height function sampled by points. Since we are working on drums, we approximate them with cylinders. The image space is mapped on the cylinder as u, v coordinates: along the cylinder axis, and in a polar angle. The polar radius of a point defines its height in the image. We project all the points of the drum in image space, and if the resolution of the image is not too large, we have no undersampling issues. To fix the remaining undersampling issues, we generate missing samples on the cylindrical coordinates grid as the average of the non missing neighbors. This technique behaves well in practice and does not change the original data.

5.2. Illumination

For the illumination, we consider this height field as a terrain, and from each point in the image, we compute an approximation of the portion of visible sky. We obtain, up to quantization error, the exact amount of lighting a terrain would have with a white sky. However, in our case, there is another approximation due to the cylindrical projection. This technique is thus biaised, but proves to be very efficient for illustration purposes.

To compute the portion of visible sky, we cast rays in 2D: around the points, in a given number of directions (32 or 40 are typical). We cast a ray in the image storing the maximal angle of the horizon. The tangent of the angle is given by the height difference divided by the distance - see Fig. 8.

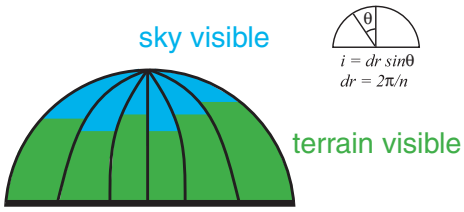

Figure 8: *Computation of the visible skydome from the horizon of a 2D height function.*

Once this value is computed for each direction, we compute the solid angle corresponding to the visible sky represented by the slice of sphere (for a given 2D direction) cut at the highest angle found in that direction. This results in a scalar value between 0 for nothing visible (or almost zero for a very steep hole), and 2π for a point on the top of a hill.

These illustrations give a very good impression of the relief of the object, even for very fine details (see Fig. 9, 10, 11).

5.3. Skydome Rendering in 3D

In the previous section we have presented an algorithm to compute skydome rendering of a *développée* considered as a terrain. In this section, we present an algorithm to efficiently compute the visible portion of the sky on a point cloud in 3D. This skydome approach, also called Ambient Occlusion, determines the portion of the environment visible from each point. The algorithm processes the data for each quantized direction independently. For each direction, we assume that the point cloud is lit by a parallel light source. We slice the point cloud along the main direction of this light direction using the octree structure. Then we propagate an occlusion mask from slice to slice which indicates whether each octree cell is transparent or not, according to the number of points included in it. Therefore we are able to determine if each octree cell of the next slice is lit or not.

The main benefit of this approach is that the propagation step is only dependent on the direction of the light. The occlusion of a cell in the mask thus only depends only on the nearest neighbors in the light direction. Accumulating visibility (i.e., the inverse of occlusion) for each direction results in the visible portion of sky.

The results of this approach have aliasing artifacts due to the cubical shape of the octree cells. We have enhanced the technique to get finer results. The occlusion of a cell is split into 9 parts, corresponding to different overlapping configurations of the cell neighbors along the light beam direction. This splitting pattern is identical for each cell given a direction. In this manner, the occlusion propagation has finer resolution, and the illumination of each point in the slice cell can be computed more precisely from this subdivision. We obtain results with substantially better quality.

Performance is not yet optimal given the huge amount of cells to be processed (about 2^{11} slices per direction, each containing 2^{22} cells, which is more than 8 billion for each direction). Moreover the occlusion of a single cell can have an influence on all the remaining slices. Thus, illumination estimation of almost each cell has to be calculated.

6. Results

As mentioned earlier, these methods have been developed in a close collaboration between the computer graphics researchers and the archeologists of the project. In particular, we discuss below the use by the archeologists of the tools developed, i.e., interactive rendering for very large point clouds, the 2D *dévelopée* rendering and the 3D skydome rendering.

6.1. Interactive rendering

The interactive rendering tool was used to examine the placement and the reconstruction of the different elements. In particular, this tool was used in the reassembly of the Omphalos on the dancers: the interactive manipulation of the 3D data of both elements gave a precise set up of the Omphalos on the caps of the 3 dancers with respect to the higher

Figure 9: *Drum 2 (2D skydome rendering of the* développée*) used in the initial search phase for docking positions.*

and lower mortises. This fine positioning had never been achieved before and it confirms the archeological hypothesis. A snapshot from the 3D viewer is shown in Fig. 12.

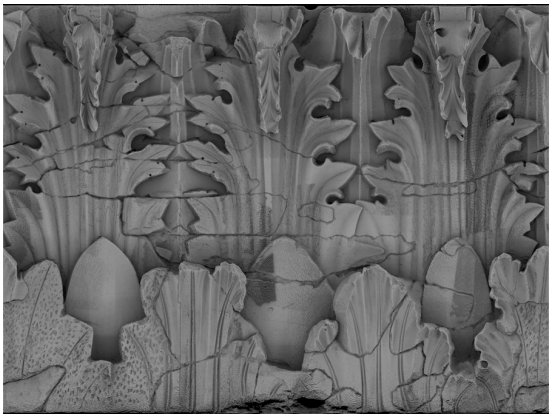

Figure 10: *Another example of a* développée *on Drum 6 and 7. Notice the rendering artifacts resulting from incorrect registration (center).*

6.2. Use of the *développée* illustration

The 2D skydome rendered *développée* has been used as a planning tool for docking and for illustration.

The fragment "Drum 1" is devastated: the docking of fragments is thus impossible due to the degree of ruin of the block. On the contrary, Drum 2 (Fig. 2) has been conserved to a large extent. We thus processed Drum 2 to create guidelines used for the docking of the fragments of Drum 1. The axis of Drum 2 has been computed by least square fitting on the 12 million points corresponding to the leading cylinder followed by the sculptors, and the guiding lines of the 48 grooves were modeled parallel to the axis.

The *développée* of Drum 2, shown in Fig. 9 was used to search for the possible positions of Drum 1 blocks. Then, referring to the guiding skeleton of Drum 2, we docked the cloud of points of each fragment of Drum 1.

As mentioned earlier, skydome rendering is used as an alternative to traditional drawing for illustration purposes, to be used in the forthcoming book on the monument [Mart04]. One interesting usage of this stylized rendering is also the fact that we can identify registration errors, due to noticeable rendering artifacts (centre of Fig. 10, and also in the central groove of Fig. 9).

We also show the *développée* of the Omphalos in Fig. 11. Note that the archeologists in this project consider this element to be impossible to draw using traditional methods.

6.3. 3D Skydome rendering

The final tool we have provided is the production of 3D images using skydome rendering of the visible portion of the sky. We show two examples, one for the Omphalos alone Fig. 13, and one for the Omphalos placed on the dancers in Fig. 12.

7. Conclusions

In this paper we have presented a complete point-based pipeline for the capture, display and illustration of very large scans. This specific project of the Delphi “Dancers Column” has involved a close collaboration between archeologists and computer graphics researchers. Archeologists specified two main requirements for this work: the possibility to interactively view in 3D the point data resulting from very large

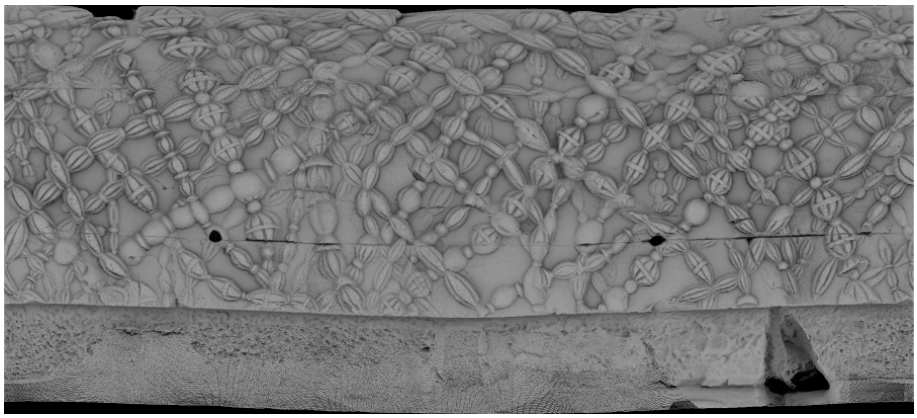

Figure 11: *The Omphalos rendering with the 2D skydome approach (*développée*).*

scans, and the ability to create stylized 2D drawings as an alternative to traditional illustration.

To achieve these goals we use a very compact representation of the data. We described how to construct this structure using out-of-core and streaming techniques. The main memory requirements for the construction of the structures is thus limited and can be performed on small, standard configurations (such as those available on the field when scanning for example). We use the rendering approach of [DD04], which, thanks to its flexibility, allows interactive viewing of the compacted data, despite the overwhelming size of the scans.

We have also presented a new stylized rendering for archeological fragments inspired from traditional *développées* in archeology. It unfolds the fragments using a cylindrical projection and then performs a skydome rendering on the resulting 2D height-field. A related technique was also presented for skydome rendering in 3D.

The techniques presented have been used successfully by the archeologists in our project. In particular the interactive 3D rendering helps in the understanding of the 3D structure of the fragments and as a tool for their fine-positioning, the *développée* was used as a planning tool in the preparation phase for docking of fragments and also as a replacement for traditional archeological illustration.

Other uses of these tools are currently under way in the context of this project, to improve the illustration and interactive viewing.

8. Acknowledgements

Electricité de France sponsored this project for the benefit of Ecole Française d'Athènes by offering its technical support and financing the partnerships. Electricité de France designed the project and managed the different partnerships under the leadership of Guillaume Thibault, whom the authors wish to thank for all his help and support. The Insight team led by K. Cain was selected for the survey and the registration of the data. We thank Ecole Française d'Athènes for having actively taken part in the realization of the project, particularly its director, Dominique Mulliez, and Dominique Braustein, curator, who supervised the handling of the elements. We also thank the Greek authorities of the Museum of Delphi who gave acces to the rooms despite the museum being closed for work, particularly his director, Rosina Kolonia and her assistant, Elena Partida. Alan Chalmers from the Department of Computer Science at University of Bristol kindly provided a MINOLTA VI910 scanner. Two students of his unit contributed actively in the scanning and processing crew. Paolo Cignoni from Visual Computing Lab of the Institute of Information Science and Technologies of the Italian National Research Council contributed to the registration of the 3D frames. We thank Jacques Droulez from Collège de France (Laboratoire de Physiologie de la Perception et de l'Action) for useful discussions about the limits of human vision and the encoding of normal directions to take these limits into account. This research was partly funded by the French Ministry of Research ACI MD "SHOW".

References

[ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (jan-mar 2003), 3–15. 6

[AGP*04] ALEXA M., GROSS M., PAULY M., PFISTER H., STAMMINGER M., ZWICKER M.:, August 2004. Siggraph Course on Point-Based Graphics. 3

[BCS01] BORGO R., CIGNONI P., SCOPIGNO R.: An easy-to-use visualization system for huge cultural heritage meshes. In *Proc. of VAST'01* (2001). 3

[BWK02] BOTSCH M., WIRATANAYA A., KOBBELT L.: Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th*

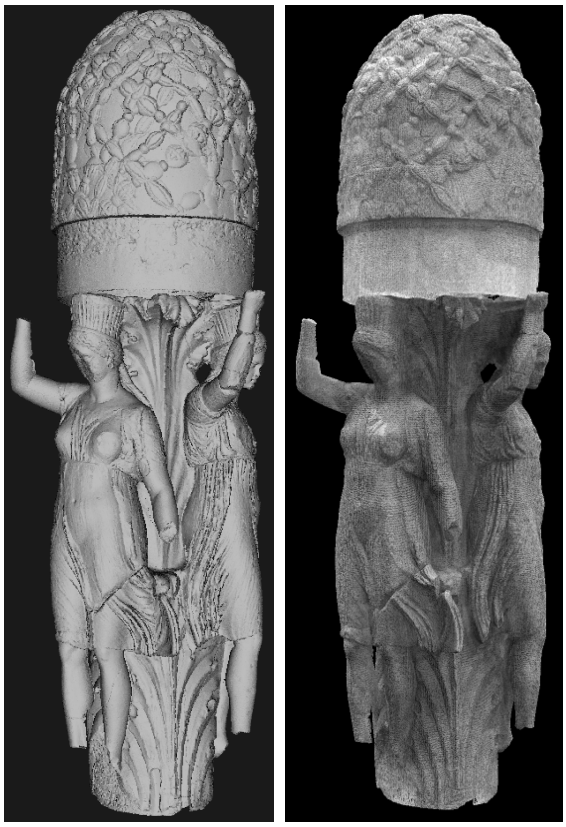

Figure 12: *Left: Interactive rendering using a local shading model, used in the set up of Omphalos on the caps of the 3 dancers. Right: The same scene using 3D visible sky portion rendering.*

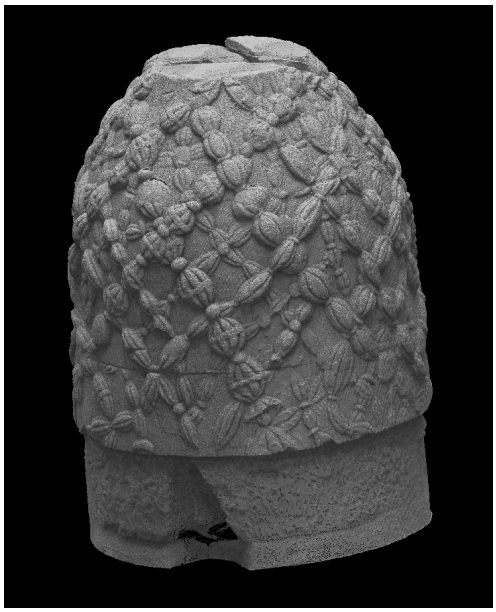

Figure 13: *The Omphalos rendering with the 3D skydome approach.*

Eurographics Workshop on Rendering (2002). 3, 5

[DD04] DUGUET F., DRETTAKIS G.: Flexible point-based rendering on mobile devices. *IEEE Computer Graphics and Applications 24*, 4 (July-August 2004). 3, 4, 5, 6, 9

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital michelangelo project: 3d scanning of large statues. In *Proc. SIGGRAPH 2000* (2000), pp. 131–144. 2

[Mart04] MARTINEZ J.-L.: *La Colonne d'Acanthe de Delphes*, vol. IV. Collection des Fouilles de Delphes, (to appear). 2, 8

[Max88] MAX N. L.: Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer 4*, 2 (July 1988), 109–117. 3

[RD03] ROUSSOU M., DRETTAKIS G.: Photorealism and non-photorealism in virtual heritage representation. In *VAST 2003 and First Eurographics Workshop on Graphics and Cultural Heritage* (November 2003), Chalmers A., Arnold D., Niccolucci F., (Eds.), Eurographics. 4

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: a multiresolution point rendering system for large meshes. In *Proc. SIGGRAPH 2000* (2000), pp. 343–352. 3

[SC00] SLOAN P. J., COHEN M. F.: Hardware accelerated horizon mapping. In *Rendering Techniques 00* (2000), pp. 281–286. 3

[SD01] STAMMINGER M., DRETTAKIS G.: Interactive sampling and rendering for complex and procedural geometry. In *Rendering Techniques 2001* (2001), Myskowski K., Gortler S., (Eds.), 12th Eurographics workshop on Rendering, Eurographics, Springer Verlag. 3

[Ste98] STEWART A. J.: Fast Horizon Computation at All Points of a Terrain With Visibility and Shading Applications. *IEEE Transactions on Visualization and Computer Graphics 4*, 1 (Jan. 1998). 3