Depth Enhanced Panoramas

Gleb Bahmutov, Voicu Popescu, Mihai Mudure, Elisha Sacks
Purdue University, Indiana, USA

Abstract
Depth enhanced panoramas are a practical solution to the difficult problem of inside-looking-out modeling. Depth enhanced panoramas extend color panoramas to support viewpoint translation, while retaining their speed, convenience, and low cost. Depth enhanced panoramas are built incrementally from same-center-of-projection dense-color and sparse-depth frames that are acquired, registered, and merged at the rate of 5 frames per second. The evolving depth enhanced panorama is rendered continually to provide immediate operator feedback. The viewpoint translation range is increased by combining multiple depth enhanced panoramas in real time. Depth enhanced panoramas are combined using either a splat-based, disconnected representation, or, at pixel level, using a GPU fragment program. Depth enhanced panoramas are built in minutes with $3,000 hardware and support photorealistic rendering of complex room-sized environments.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling.

1. Introduction

Interactive visualization of real-world scenes has diverse applications. Historical building visualization benefits cultural heritage preservation. Visualization of crime and accident scenes facilitates forensic investigation. Web-based marketing of houses, hotels, and other buildings calls for interactive rendering of actual rooms. Detailed models of the interior of actual buildings with hundreds of rooms are invaluable for virtual training of emergency response personnel. Such models are also important for physical simulations. For example, the propagation of an air-borne contaminant can be more accurately simulated using a model that describes the rooms down to the level of their individual furniture configuration. Another example is acquiring the scene after a crash test experiment and using the model to improve the algorithms and fine-tune the parameters used in simulation codes.

The challenge is to build digital models of complex scenes that support interactive, realistic visualization. Manual modeling is unacceptably labor intensive. The natural alternative is automated modeling where the scene color and geometry are sampled using an acquisition device. Many automated modeling techniques have been developed, but none offers a complete solution to real-world scene modeling. Systems that build high-quality models are slow (depth from stereo, laser rangefinding), require expensive, bulky equipment (laser rangefinding), or require manual intervention (image-based editing, view morphing, tour into the picture). Systems that do not suffer from these disadvantages build simplified scene models (color panoramas) that must be viewed from preferred locations.

We describe an interactive method of modeling and visualizing room-size indoor scenes that is fast, simple, and

Figure 1: *The room in the two top images and the room in the bottom image are each modeled with a single depth enhanced panorama. The depth enhanced panoramas have 299,820/169,083 triangles and each was acquired under 30 minutes with a $3,000 device.*

inexpensive (see Figure 1). We use a structured light acquisition device that consists of a video camera enhanced with a laser system [PSB04]. The device acquires frames of dense color (720x480) and sparse depth (7x7) at 15 frames

per second. The acquisition device is mounted in a bracket that allows it to pan and tilt around the camera's center of projection. As the operator sweeps the scene, the frames are registered in world coordinates and are merged into an evolving scene model, called a depth enhanced panorama (DEP). The model is displayed continually to provide immediate feedback to the operator.

DEPs remove the fundamental limitation of color panoramas by supporting viewpoint translation, yet retain their speed, convenience, and low cost. Like color panoramas, DEPs take advantage of the descriptive power of high-resolution color. When viewed from the acquisition viewpoint, a DEP is equivalent to a color panorama. When the view translates away from the acquisition viewpoint, the depth samples of the DEP exhibit motion parallax revealing the 3D shape of the scene. For many applications, and these certainly include the numerous applications where color panoramas are presently used, a single DEP captures the scene well enough. More detailed models are obtained by using several DEPs of the same scene, and we describe techniques for combining DEPs in real time.

Automated modeling is traditionally a computer vision problem. In the last decade automated modeling has also received considerable attention from computer graphics researchers that regard it as a promising way of further approaching the goal of photorealistic rendering at interactive rates.

We distinguish between *outside-looking-in modeling*, when the scene is acquired from outside its volume, and *inside-looking-out modeling*, when the acquisition device is inside and surrounded by the scene. In the outside-looking-in case a good coverage of the scene is obtained by orbiting the acquisition device around the scene or by spinning the scene in front of the acquisition device. Moreover the lighting of the scene can be controlled and depth needs only to be measured over a small, fixed range. In inside-looking-out modeling achieving good coverage is more challenging, and the need of measuring depth and light intensity over wider ranges complicates acquisition.

However, applications where the user explores the scene from within, thus requiring inside looking out modeling are the ones that would mostly benefit from a photorealistic model of the scene. While an object is well described by a set of digital photographs, for example the interior of a building is best revealed if the user can freely wander inside with the support of high-quality interactive rendering. Outside-looking-in modeling is of course very useful for applications that go beyond showing the scene from a novel view, such as describing the use of an object, reverse engineering, fast search and comparison in large databases of acquired objects, and creation of novel objects by interpolation.

This paper describes a novel technique for modeling and rendering large, room sized environments, and we will limit the prior work discussion to systems suitable for inside-looking-out modeling.

2. Prior work

Ray databases: QuickTime VR panoramas [Shen95] are 2D ray databases that store a dense sampling of the rays passing through one point. They are constructed by stitching together same-center-of-projection images. They support viewing the scene from this point in any desired direction. Panoramas have the advantages of rapid, inexpensive acquisition and of interactive photo realistic rendering. The disadvantage of panoramas is that they do not support view translations; this deprives the user of motion parallax, which is an important cue in 3D scene exploration. Panoramas have probably the largest interactive-rendering-quality/cost ratio, claim supported by their widespread use. On a cost versus quality graph, panoramas are at the bottom left (Figure 2).

Light fields [LH96, GGSC96] are 4D ray databases that allow a scene to be viewed from anywhere in the ray space. An advantage of light field rendering is support for view dependent effects, such as reflection [LH96, GGSC96, Wood00], occlusion, and refraction [Matusik02]. Light fields are constructed from a large set of registered photographs. Acquiring and registering the photographs is challenging. Another disadvantage is that interpolating a ray of the desired view can be done accurately only if an approximate geometric model of the scene is available [GGSC96] or when the user selects the focal plane [Isaksen00]. The main disadvantage is that the ray database needed to explore a scene of the size of a room is impractically large. In our chart light fields are at the top right for handling reflective objects and objects with extremely complex geometry, but having prohibitive cost for large scenes and large viewing volumes.

Manual depth: Another solution to the depth acquisition problem is manual geometry data entry. An example is the Facade architectural modeling system in which the user creates a coarse geometric model of the scene that is texture mapped with photographs [DTM96]. The geometric part of the hybrid geometry-image-based representation is created from user input in [HH02]. In view morphing [SD96], the user specifies depth in the form of correspondences between reference images. Another example is image-based editing [AHA97, OCDD01], which builds 3D models by segmenting images into sprites that are mapped to separate planes. Manual depth systems take advantage of the users' knowledge of the scene, which allows them to maximize the 3D effect while minimizing the amount of depth data. The disadvantage of the approach is that manual geometry specification is slow and difficult. We place the manual depth methods to the right of panoramas since they allow translation, but that comes at a higher cost.
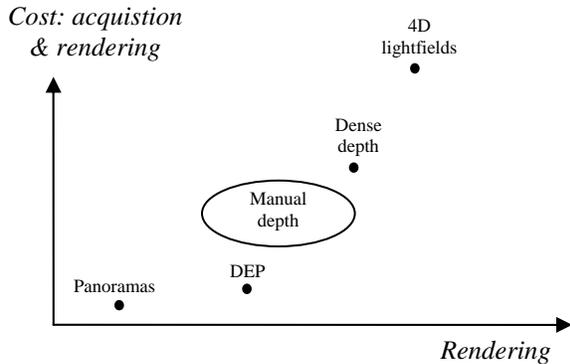
Figure 2: *Cost/quality tradeoff for various automated modeling techniques.*

Dense depth: Depth from stereo, structured-light laser rangefinding, and time-of-flight laser rangefinding technologies acquire dense, accurate depth maps that can be converted into high-quality models. Examples include the digitization of Michelangelo's statues [Levoy00, BMMR02], of Jefferson's Monticello [WHL*03], of cultural treasures of Ancient Egypt [FEE*03], of the Parthenon [STY*03], and of the ancient city of Sagalassos [PV02, PVAD01]. McMillan and Bishop [MB95] describe modeling real-world scenes using *depth images*, which are panoramas with per-pixel depth inferred by stereo matching. The depth images are transformed efficiently to the desired view by 3D warping. Dense depth methods produce good geometry definition but suffer from long per-view acquisition times and high equipment costs.

The goal of the work described in this paper is to bridge the gap between high-quality high-cost methods and approximate, low-cost methods, by improving the models that can be obtained without a great time and equipment cost increase. The approach taken is interactive modeling. The scene model is built incrementally from small pieces that are acquired, registered and integrated in rapid succession under the operator's supervision, which is made possible by real-time feedback. Interactive modeling is a recent but rapidly growing subfield of automated modeling.

Interactive modeling: Rusinkiewicz et al. [RHL02] present an object modeling system based on structured light. The evolving model is constructed in real time and is rendered to provide immediate feedback to the operator. The system does not acquire color, and is limited to small objects. A similar system is proposed by Koninckx [KGV03] where moving or deformable objects are captured in real time. The disadvantages of their system are limited acquisition range due to the fixed camera and projector configuration and the need for strict lighting control. Despite their shortcomings, both systems demonstrate the advantages of interactive modeling.

ModelCamera

We use a structured-light acquisition device called the ModelCamera. An earlier prototype was described in [PSB03]; the current prototype (Figure 3) was described in [PSB04]; we describe it here briefly for completeness.

The ModelCamera consists of a mid-level video camera and a commodity laser system [Stockeryale]. It weighs 1kg and costs $3000 to build. The laser is eye safe (class IIIa), yet is powerful enough to produce bright dots in indoor scenes. The camera is connected to a PC by a FireWire interface. Depth samples are obtained by undistorting the frame, finding the laser dots, and computing their corresponding 3D positions. Each dot is restricted to an epipolar line because the lasers are fixed with respect to the camera. The lasers are configured to make the epipolar lines disjoint, which prevents ambiguity in dot/laser assignment. The line is truncated to the segment

Figure 3: *ModelCamera mounted in parallax-free pan-tilt bracket.*

that contains the usable depth range of 50cm-300cm. The segment is searched for an intensity peak that passes 2D symmetry tests. We exploit coherent camera motion by starting the search at the dot from the previous frame. The 3D position of the dot is calculated by triangulation between the optical ray through the dot and the laser beam.

The system acquires 720x480 video frames enhanced with 49 evenly spaced depth samples. The depth data is intrinsically registered with the color data, since depth is inferred from color. This is an advantage over systems that acquire depth and color from separate devices, hence must coregister the data. Dot detection takes 5ms per frame. The detection error is 0.5 pixels, which implies a depth accuracy of 0.1cm at 50cm, 0.4cm at 100 cm, and 1.2cm at 200cm.

The ModelCamera operates in two fundamental modes. *Structured scenes* are acquired freehand [PSB03]. Using surface continuity the depth-enhanced frames are registered and merged into an evolving model which consists of depth images. Unstructured scenes are modeled with DEPs which are acquired using a parallax-free camera bracket (Figure 3). The idea of depth enhanced panoramas was briefly introduced in [PSB04]. The present paper describes DEPs in detail, introduces a disconnected representation for DEPs, and presents modeling and rendering a scene from multiple viewpoints using DEPs that are registered and combined in real time.

3. DEP construction

A DEP consists of a color cube map enhanced with depth samples, and is constructed by registering and merging a sequence of dense color and sparse depth frames. Registration transforms the current frame data from camera coordinates to world coordinates. Since the frames share a common center of projection, they can be registered using only the color data, in the same way that images are stitched together to form color panoramas. Each new frame is registered against the faces of the cube map with which it overlaps.

We have developed a fast registration algorithm that minimizes a color error function whose arguments are the

pan and tilt angles. In order to allow the operator to start from any tilt position, we find the starting tilt angle on the fly at the beginning of each sequence. The operator is required to pan, and only pan, the camera for at least 15 degrees at the beginning of the sequence. This way the tilt angle is not changed from its initial value. The frames are registered using color as described below. The tilt angle is computed from the cone described by the camera's view direction. 15 degrees are sufficient for computing the initial tilt angle robustly. Once the initial tilt angle has been established the operator is informed that the camera can also be tilted.

A frame is registered by minimizing its color error. The color error of a frame is defined as the sum of the color errors of each pixel considered. The error of a pixel is the RGB distance between its color and the color where it projects in the cube map. Even small camera motions produce rapid, erratic changes in color error. We reduce the variability and the noise by convolving the frames with an 11x11 raised cosine filter.

We then select a registration pixel pattern in the current frame. The pattern consists of horizontal and vertical segments that exhibit considerable color variation. The projection equations for a pixel (u, v) are two fractions of linear expressions in u and v, with the same denominator, that give the u' and v' coordinates in the cube map face. The pixels of a segment share the same row or column and thus can be projected onto the cube map faces with an amortized cost of 3 additions (to advance the linear expressions of the two numerators, and the shared linear expression of the denominator) and 2 divisions (to compute the two fractions). We minimize the sum of the square of the pixel errors by the downhill simplex method. The dot pixels are excluded because their color comes from the lasers, rather than from the scene. The simplex method does not require derivatives, which are expensive to compute.

Each registered frame with its depth samples is added to the DEP. The frame's color is added to the color data already accumulated in the color cube. The faces of the color cube are divided into tiles for efficiency. The current frame updates only the tiles that fall within its field of view and are not yet filled. The depth samples are stored separately for each face to which they project. For each face a quadtree is created to accelerate the access to the closest neighbors for a given depth sample.

Registration takes 150ms per frame and merging takes 50ms per frame, so the modeling rate is 5 frames per second. When there is sufficient color variation in the scene, the segments and the cube map faces are downsampled by a factor of up to 10, which accelerates DEP construction to 10 frames per second. The registration algorithm fails once in 100-300 frames on average. The operator easily regains registration by aligning the camera view with the last registered frame.

4. DEP visualization

During the acquisition the operator can inspect the 3D model of the already scanned portion of the scene without stopping / restarting the acquisition. We have developed two DEP visualization methods that produce high-quality images of the evolving model of the scene at interactive rates. Displaying the model as it is being created is not only useful to the operator but also benefits applications where remotely located users have the ability to inspect the acquired model without delay.

Disconnected representation

A disconnected visualization method for DEPs is similar to the splatting techniques of point-based modeling and rendering: QSplats [SL00], surfels [PZVG00], and forward rasterization [Popescu00]. None of these methods applies, since DEPs are sparsely populated with depth samples. Instead, we generate a texture-mapped square splat for each depth sample. The splat's size and normal are derived from the neighboring depth samples, located quickly using the quadtree data structure. The neighbors are triangulated and the normals of the triangles are averaged to obtain the splat normal. The splat size is an average distance from the depth sample to its neighbors. This fills most of the gaps that would otherwise appear due to the sparse set of depth samples in the DEP. The splats are texture mapped using the cube map faces (Figure 4 right column).

Connected representation

An alternative to splatting is the connected representation of the DEP. It is built by triangulating in 2D
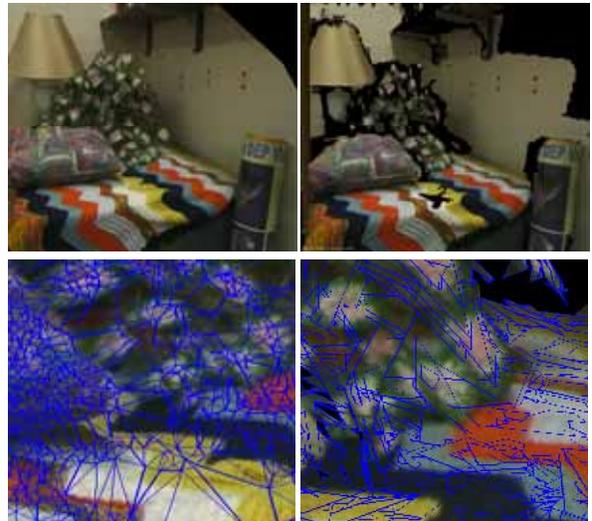

Figure 4: *(Top left) DEP splatting. (Bottom left) Magnified fragment with highlighted splat boundaries.*

(Top right) DEP shown as a connected triangle mesh. (Bottom right) Magnified fragment with superimposed wireframe

Figure 5: *Depth samples are triangulated in 2D on the face of the panorama (orange) then the inferred connectivity is used to make the 3D mesh (blue).*

the depth samples projected onto the faces of the cube map. A 3D triangle mesh is created by applying this connectivity data to the corresponding 3D depth samples (Figure 5). The 3D triangle mesh is texture-mapped with the cube map faces (Figure 4 left column).

During the acquisition, the 2D mesh is triangulated incrementally to accommodate the depth samples of the newly integrated frame. We use a Delaunay tree with logarithmic expected insertion time [DMT92, Devillers 92, BT93]. The implementation was obtained from [Delaunay]. The videos on our website [ModelCamera] illustrate real-time incremental triangulation of DEPs.

Connected representation as a tool for point filtering

False positives in dot detection cause large distortions in the 3D model. We avoid false positives relying on frame to frame coherence. A 3D point is added to the depth enhanced panorama only if it was at roughly the same 3D position for several frames. The number of frames considered is dynamically adapted according to the distance between the positions of points in consecutive frames.

For very fragmented surfaces, such as a plant, this filtering approach is too conservative and rejects too many valid points. When scanning the plant on the table shown in Figure 4, less than 30% of the triangulated depth samples were admitted into the DEP using three buffer frames and a maximum allowable point movement of 2.0 cm.

In order to reduce the number of good points rejected we use a second chance strategy for the rejected depth samples that takes advantage of the evolving connected DEP representation. The rejected points are stored in a temporary buffer. As the evolving triangle mesh is visualized from the acquisition point, the OpenGL z buffer is read back periodically (every k=50 registered frames in our experiments). A 3D point rejected by the frame coherence filter is projected onto the image plane. The z of the projected point and the z buffer value at the projection location are used to infer the distance from the point to the
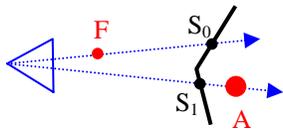

Figure 6: *Since the length of the segment FS_0/AS_1 is more/less than the threshold the point F/A is accepted/rejected.*

surface along the viewing ray (Figure 6). If the evolving surface is not covering this pixel, the 3D point is left in the temporary buffer. Otherwise, if the 3D distance along the viewing ray to the surface is less than a set threshold (we use 1.0 cm), it is added to the DEP. If the 3D distance is larger than the set threshold, the 3D point is rejected.

The second chance method allowed us to increase the percentage of admitted depth samples to 80% percent of the total number of triangulated 3D points. The additional computation cost is low and no false positives were admitted as confirmed by the lack of outliers in the model. The DEP shown in Figure 4 was acquired in 318 seconds, contains 44,996 depth samples, and thus on average 141 depth samples per second were admitted to the DEP.

5. Multiple viewpoints

If the desired view is close to the acquisition viewpoint, a single DEP produces high-quality images of the scene. If the desired view is considerably different from the DEP acquisition view, the image quality degrades because of missing and undersampled surfaces (Figure 8 middle row). A wider range of views is supported by acquiring, registering and displaying several DEPs of the scene. The operator builds the first DEP as before, examines it for missing or poorly sampled surfaces, moves the ModelCamera to a second viewpoint, and starts building the second DEP. Once sufficient surfaces are acquired, the second DEP is registered with the first using three operator-specified point correspondences between the two DEPs. The system computes the rigid camera motion between the two DEPs, the acquisition of the second DEP resumes, and the two DEPs are visualized together.

Disconnected representation

The disconnected representation supports multiple DEP visualization without modification. The already completed DEPs and the evolving DEP are continually rendered independently from each other in the splatting mode to guide the operator in completing the model. See the accompanying video.

Connected representation

To achieve higher quality visualization than a disconnected representation allows we developed a runtime per pixel sample selection algorithm to combine the best parts of several DEPs. The straightforward z-buffered rendering of the multiple DEPs does not produce realistic novel views, since better sampled surfaces are often obscured by the worse sampled ones. To solve this problem, we wrote a GPU fragment program that selects the input sample for every pixel based on z and sampling rate (Figure 8).
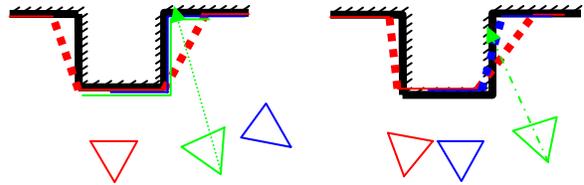

Figure 7: *Left: two input DEPs (red and blue) of a scene are combined to produce a novel view (green). The best sample for the desired view ray shown is the sample coming from the second DEP, although the sample coming from the first DEP is closer.*

Right: The desired view ray intersects undersampled triangles in all DEPs. We select the farthest sample, in this case the one coming from the second DEP. The farther sample invalidates the triangle coming from the first DEP, and indicates that the second DEP approximates the scene more closely.

First, during a preprocessing step, we compute the sampling rate of each triangle as the inverse of the average length of its sides, normalized to 0..1 range. Triangles with sampling rates below a given threshold *SR* are labeled as

undersampled. *SR* is established experimentally for each scene.

During runtime we select the three DEPs with acquisition viewpoints closest to the desired viewpoint. Each input DEP is rendered from the desired view into a separate high-resolution 2048x2048 pixel OpenGL color and z buffer. The sampling rate of the triangle is stored in the alpha channel. The separate color and z buffers are bound as texture maps and combined on per pixel basis using the fragment program. If a pixel is covered in at least one DEP by a triangle that is not undersampled, the closest such triangle is used If an output pixel is covered by undersampled triangles in all of the DEPs, the algorithm selects the farthest undersampled surface (Figure 7).

The rendering frame rate depends on two main factors: the number of primitives in each input DEP and the target frame buffer resolution. The rendering of the static input DEPs is optimized by using the OpenGL compiled lists. The per pixel merging of the individual OpenGL color and depth buffers is sped up by using WGL_ARB_pbuffer and GL_ARB_multitexture extensions [OpenGL] which allow drawing into and combining multiple rendering contexts on the graphics card without transfers of the pixel buffers to/from main memory. We have achieved 5 fps rendering rate using the nVidia Quattro FX 3000 graphics card for a 512x512 output frame buffer and three input DEPs with more than 40K triangles each. The computation of the triangle sampling rate as preprocess took less than a second.

We have observed visual artifacts in the undersampled regions, due to very coarse geometry of the DEPs in these areas. We believe that the ability to inspect the results immediately, as we have achieved, allows the operator to identify the undersampled regions, and improve the model appearance by scanning the scene from new locations.

6. Conclusions

Depth enhanced panoramas have the advantages of the traditional color panoramas: DEPs can be acquired quickly and inexpensively, yet overcome their fundamental limitation by allowing view point translation. DEPs have a good quality/cost ratio and cover a void in the quality-cost tradeoff space. They have the potential to enable novel applications of automated modeling.

DEPs prove the power of interactive modeling from dense color and sparse depth. At the rate of 5 fps, the ModelCamera acquires 100,000 depth samples in 10 minutes of continuous operation. The operator maximizes the impact of the depth samples by scanning in greater detail the parts of the scene with complex geometry and by avoiding redundant scanning of flat regions. The disconnected and connected visualization methods we have developed allow the operator to inspect the evolving or finished model of the scene at the interactive rendering rates. We devised a method to use the evolving connected representation to avoid rejection of too many depth samples during the acquisition. We extended both disconnected and connected representations to handle multiple DEPs of the scene, significantly increasing the range of the positions that generate high quality novel views of the scene.

We are designing a new prototype with a custom laser system that is brighter and acquires 121 depth samples per frame. Our goal is to model one room in one hour and entire buildings in a single day by scanning in parallel.

7. References

[AHA97] Anjyo, K., Horry, Y., and Arai, K. "Tour into the Picture" Proc. SIGGRAPH '97 pp. 225-232.

[BMMR02] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, G. Taubin. Building a Digital Model of Michelangelo's Florentine Pieta'. IEEE Computer Graphics & Applications, Jan/Feb. 2002, 22(1), pp. 59-67.

[BT93] J.-D. Boissonnat and M. Teillaud, On the randomized construction of the Delaunay tree, Theoret. Comput. Sci. 112:339--354, 1993.

[Shen95] S. Chen, Quicktime VR - An Image-Based Approach to Virtual Environment Navigation, Proc. SIGGRAPH 95, 29-38 (1995).

[DTM96] P. Debevec, C. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image Based Approach. Proc. SIGGRAPH '96, 11-20 (1996).

[Delaunay] http://www.csit.fsu.edu/~burkardt/cpp_src/delaunay_tree_2d/delaunay_tree_2d.html

[DMT92] O. Devillers, S. Meiser and M. Teillaud, Fully dynamic Delaunay triangulation in logarithmic expected time per operation, Comput. Geom. Theory Appl. 2(2):55-80, 1992.

[D92] O. Devillers, Robust and efficient implementation of the Delaunay tree, INRIA Research Report 1619, 1992.

[FEE*03] M. Farouk, I. El-Rifai, S. El-Tayar, H. El-Shishiny, M. Hosny, M. El-Rayes, J. Gomes, F. Giordano, H. Rushmeier, F. Bernardini, and K. Magerlein, "Scanning and Processing 3D Objects for Web Display", 4th International Conference on 3D Digital Imaging and Modeling (3DIM '03), Banff, Alberta, October 2003.

[GGSC96] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. Proc. of SIGGRAPH 96, 43-54 (1996).

[HH02] E. Hidalgo and R. J. Hubbold. Hybrid geometric-image-based-rendering. Proceedings of Eurographics 2002, Computer Graphics Forum, 21(3):471-482, September 2002.

[Isaksen00] Isaksen, A. et al. Dynamically Reparameterized Light Fields, Proc SIGGRAPH 2000.

[KGV03] T. P. Koninckx, A. Griesser, and L. Van Gool, Real-Time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light. Proceedings of Fourth International Conference on 3D Digital Imaging and Modeling 2003, pp. 293-301.

[Levoy00] M. Levoy et al. The Digital Michelangelo Project: 3D Scanning of Large Statues, Proc. ACM SIGGRAPH, 2000.

[LH96] M. Levoy, and P. Hanrahan. Light Field Rendering. Proc. of SIGGRAPH 96, 31-42 (1996).

[Matusik02] Matusik, W. et al. Image-Based 3D Photography using Opacity Hulls. Proc. Of SIGGRAPH '02.

[MB95] McMillan, L., and Bishop G. Plenoptic Modeling: Image-Based Rendering. In proceedings of SIGGRAPH 1995.

[ModelCamera] http://www.cs.purdue.edu/cgvlab/ModelCamera

[OCDD01] Byong Mok Oh, Max Chen, Julie Dorsey, and Fredo Durand. Image-Based Modeling and Photo-Editing Proceedings SIGGRAPH 2001

[PV02] M. Pollefeys and L. Van Gool. From Images to 3D Models, Communications of the ACM, July 2002/Vol. 45, No. 7, pp.50-55.

[PVAD01] M. Pollefeys, L. Van Gool, I. Akkermans, D. De Becker, "A Guided Tour to Virtual Sagalassos", Proc. VAST2001 (Virtual Reality, Archaeology, and Cultural Heritage)

[PZVG00] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface Elements as Rendering Primitives. Proc. of SIGGRAPH 2000, 335-342.

[PSB03] V. Popescu, E. Sacks, and G. Bahmutov. The ModelCamera: A Hand-Held Device for Interactive Modeling. Proc. Fourth International Conference on Digital Imaging and Modeling, Banff, 2003.

[PSB04] V. Popescu, E. Sacks and G. Bahmutov "*Interactive Modeling from Dense Color and Sparse Depth*" (3DPTV 2004)

[Popescu00] V. Popescu et al. The WarpEngine: An Architecture for the Post-Polygonal Age. Proc. of SIGGRAPH 2000.

[SL00] S. Rusinkiewicz, M. Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. Proc. SIGGRAPH 2000.

[RHL02] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-Time 3D Model Acquisition. Proc. SIGGRAPH 2002.

[SD96] S. M. Seitz and C. R. Dyer. View Morphing Proc. SIGGRAPH 96, 1996, 21-30.

[OpenGL] http://oss.sgi.com/projects/ogl-sample/registry/ OpenGL extensions specification.

[Stockeryale] http://www.stockeryale.com/

[STY*03] Jessi Stumpfel, Christopher Tchou, Nathan Yun, Philippe Martinez, Timothy Hawkins, Andrew Jones, Brian Emerson, Paul Debevec. Digital Reunification of the Parthenon and its Sculptures, 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, Brighton, UK, 2003.

[WHL*03] Nathaniel Williams, Chad Hantak, Kok-Lim Low, John Thomas, Kurtis Keller, Lars Nyland, David Luebke, and Anselmo Lastra. Monticello Through the Window. Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST 2003), Brighton, UK (November 2003).

[Wood00] Wood, D.N. et al. 2000. "*Surface light fields for 3D photography*". Proceedings, SIGGRAPH '00, ACM Press, pp.287-296.
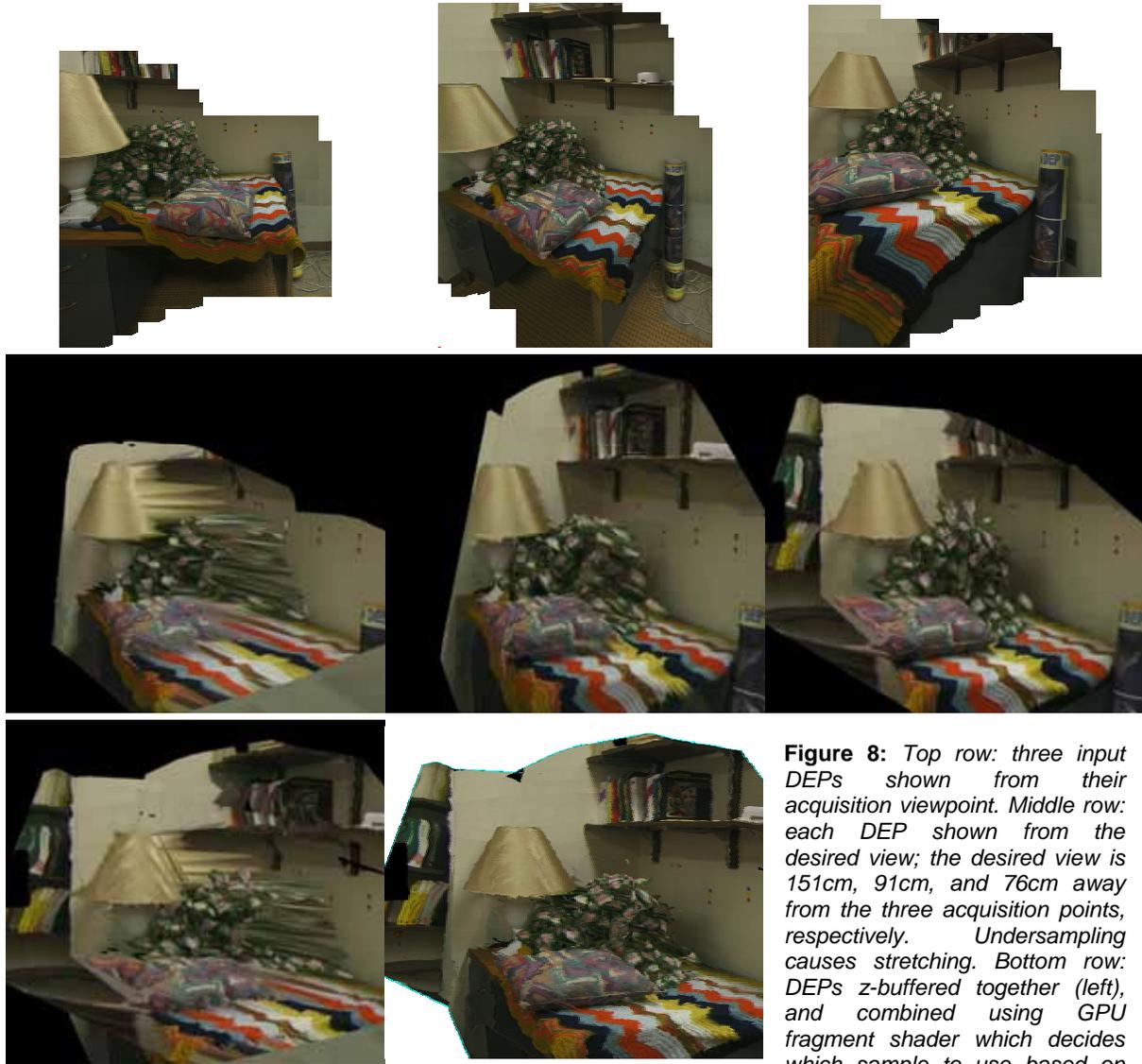
Figure 8: *Top row: three input DEPs shown from their acquisition viewpoint. Middle row: each DEP shown from the desired view; the desired view is 151cm, 91cm, and 76cm away from the three acquisition points, respectively. Undersampling causes stretching. Bottom row: DEPs z-buffered together (left), and combined using GPU fragment shader which decides which sample to use based on both depth and sampling rate (right).*