# Depth Enhanced Panoramas

Gleb Bahmutov, Voicu Popescu, Elisha Sacks
Purdue University {bahmutov, popescu, eps}@cs.purdue.edu

Figure 1 Depth enhanced panorama of a room: 100,000 triangles acquired in 30 minutes with a $3,000 device.

We describe a method for interactive modeling and visualization of room-size indoor scenes that is fast, easy, and inexpensive. We have designed a novel data acquisition device, called a ModelCamera, that consists of a video camera with an attached laser system that generates a 7x7 pattern of depth samples. The operator pans and tilts the ModelCamera around the camera's center of projection and it acquires a sequence of color and depth frames. The frames are registered in world coordinates and are merged into an evolving scene model, called a depth enhanced panorama. The model is visualized continually; the immediate feedback allows the operator to monitor the model quality and to identify missing scene surfaces. Our approach extends color panoramas to support viewpoint translation, while retaining their speed, convenience, and low cost (Figure 1). It is a practical alternative to modeling systems that provide complete models at a high cost.

## DEPTH ENHANCED PANORAMA ACQUISITION

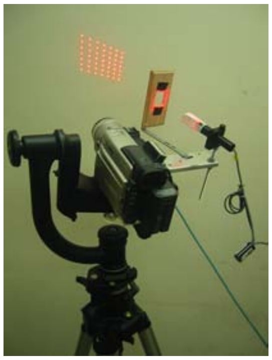
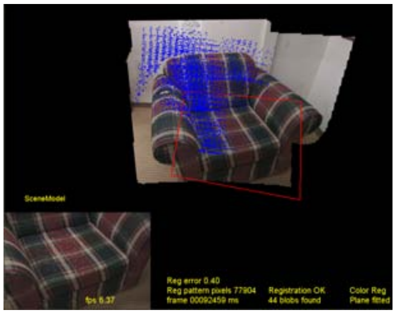


Figure 2 ModelCamera mounted in parallax-free pan-tilt bracket.

The ModelCamera (Figure 2) consists of a mid-level video camera and a commodity eye-safe laser system. It weighs 1kg and costs $3000 to build. The camera is connected to a PC (2GHz 2GB Pentium Xeon) by a FireWire interface. Depth samples are obtained by undistorting the frame, finding the laser dots, and triangulating their 3D positions. Each undistorted video frame and its sparse depth samples are registered against already filled portions of a color cube map at a rate 5 fps. The color information is added to the unfilled tiles of the cubic color map. The accumulated registered depth samples allow translation of the viewpoint away from the acquisition point, providing motion parallax. Together, the color cube map and registered depth samples form a depth enhanced panorama that can be effectively visualized from novel viewpoints during and after acquisition. The evolving model is constantly presented to the operator (Figure 3), guiding the operator to the undersampled regions. If the registration fails, the operator regains registration by matching the current frame to the last registered one, and continues scanning.



Reg error 0.40
Reg pattern pixels 77904
frame 00092459 ms
Registration OK
44 blobs found
Color Reg
Plane fitted

Figure 3 Registration feedback: current frame (*bottom left*), last registered frame (*red rectangle*), depth samples (*blue*).

## DEP VISUALIZATION

We have developed two DEP visualization methods that produce high-quality images of the scene at interactive rates during scene acquisition. The methods support real-time visualization of evolving DEP's, which is integral to interactive modeling.

### *Connected representation*

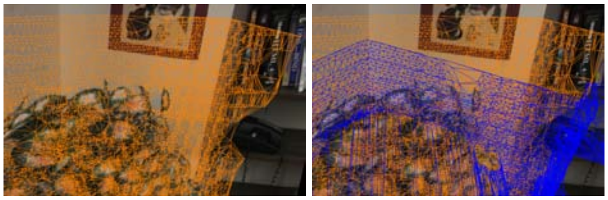


Figure 4 The depth samples are triangulated on the face of the panorama (*orange*) then the inferred connectivity is used to make the 3D mesh (*blue*).

A connected representation of the DEP is built by triangulating the projected depth samples on the faces of the cube map. A 3D triangle mesh is created by applying this connectivity data to the 3D depth samples (Figure 4). The 3D triangle mesh is texture-mapped with the cube map faces. The mesh is extended with new samples using dynamic Delaunay tree insertion [Devillers 1992].

### *Disconnected representation*

We have developed a disconnected visualization method for DEP's that is similar to the splatting techniques of point-based modeling and rendering: QSplats [Rusinkiewicz 2000] and Surfels [Pfister 2000]]. None of these methods applies, since DEP’s are sparsely populated with depth samples. Instead, we generate a texture-mapped square splat for each depth sample. The splat size and normal are derived from the neighboring depth samples. We store the neighbors in quad-trees for real-time access. Each depth sample is stored in the appropriate tree using its projection onto its face. The neighbors are triangulated and the normals of the triangles are averaged to obtain the splat normal. The splat size is derived from the distances to the neighboring depth samples. The splats are texture mapped with the cube map faces (Figure 5). The splatting visualization technique is used when the scene is modeled with multiple DEP’s.

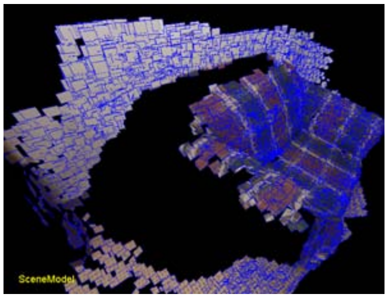


Figure 5 DEP Splatting. The splats contour is shown with the blue wireframe.

## MULTIPLE VIEWPOINTS

DEP’s work because they capture the scene from a single view point. If the desired view is close to this viewpoint, a single DEP produces high-quality visualizations of the scene. If the desired view is very different from the DEP acquisition view, the image quality degrades because of missing and undersampled surfaces (Figure 6). A wider range of views is supported by acquiring several DEP’s of the same object.

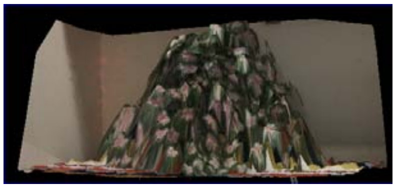


Figure 6 DEP rendered from extreme view.

The operator builds the first DEP as before, examines it for missing or poorly sampled surfaces, moves the ModelCamera to a second viewpoint, registers that viewpoint, and builds a second DEP. The Viewpoint registration is interactive: the operator specifies three pairs of corresponding scene points in the two views and the system computes the ModelCamera motion between the viewpoints. The first DEP and the evolving second DEP are visualized continually in the splatting mode to guide the operator in completing the model (see Figure 7).

## CONCLUSIONS

DEP’s have the advantages of color panoramas of fast, inexpensive acquisition, yet overcome their fundamental limitation by allowing view point translation. DEP’s 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 such as local cultural heritage preservation.

DEP’s 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.

Immediate future work plans include devising better methods for merging DEP’s. One possibility is to switch from one DEP to another according to the current desired view, similar to view dependent texture mapping. The motion parallax due to the depth samples provides a natural, approximate morph of one DEP into the next. A challenge is to alleviate the popping artifact when switching from one DEP to another. Another possibility for merging DEP’s is to merge their individual geometries. The challenge here is to combine two approximate representations into a better representation.

DEP’s are part of the ModelCamera interactive modeling system. We are designing a new prototype with a custom laser system that is brighter and acquires 100-200 depth samples per frame. In parallel with DEP’s, we will develop a freehand modeling method for *structured* scenes that contain a few smoothly varying surfaces. Our goal is to model one room in one hour and entire buildings in a single day by scanning in parallel.

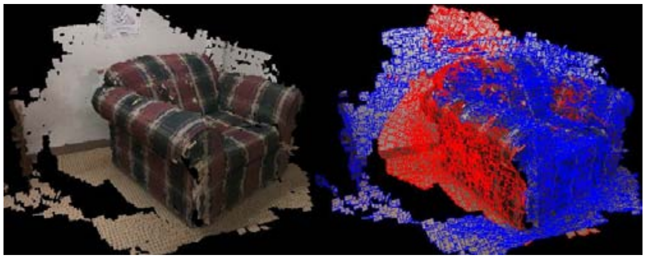


Figure 7 Model obtained by merging two DEP’s. Splats highlighted according to DEP to which they belong (right).

## ACKNOWLEDGEMENTS AND REFERENCES

We thank Rajesh Selvamani for his help with the implementation of the DEP splatting module, Chun Jia for her help with an earlier implementation of the panorama stitching algorithm, and the Purdue University Computer Science graphics group for useful discussions. This research was supported by NSF grant IIS-0082339 and CCR-0306214, and by the Purdue University Center for Graphics, Geometry, and Visualization.

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

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

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