Interactive Modeling from Dense Color and Sparse Depth

Voicu Popescu, Elisha Sacks, and Gleb Bahmutov

Computer Science Department, Purdue University

Abstract

We are developing a system for interactive modeling of real world scenes. The acquisition device consists of a video camera enhanced with an attached laser system. As the operator sweeps the scene, the device acquires dense color and sparse depth frames that are registered and merged into a point-based model. The evolving model is rendered continually to provide immediate operator feedback. The interactive modeling pipeline runs at five frames per second. We model scenes in two modes based on their geometric complexity. Scenes that contain large smooth surfaces are modeled freehand; scenes that contain small uneven surfaces are modeled using a parallax-free camera bracket.

1. Introduction

We present research in scene modeling. The task is to build digital models of natural scenes that support interactive, photorealistic rendering. Scene modeling is the bottleneck in many computer graphics applications, notably virtual training, geometric modeling for physical simulation, cultural heritage preservation, internet marketing, and gaming. Capturing complex scenes with current modeling technology is slow, difficult, and expensive. We describe an interactive modeling system that has the potential to solve these problems.

The traditional approach to modeling natural scenes is manual modeling using animation software (3dsmax, Maya). Manual modeling requires artistic talent, technical training, and a huge time investment.

The alternative is automated modeling according to the following pipeline. Color and geometry data is acquired from a few views. Color is acquired with a camera. Geometry is inferred from the color data or is measured with a depth acquisition device. The data from each view is given in a local coordinate system, so it must be registered in a common, world coordinate system. Model construction software discards redundant data, interpolates missing data, and encodes the results into a format that is suitable for rendering.

Data acquisition takes tens of minutes for each view because depth acquisition is slow (due to sequential high-resolution scanning in laser rangefinding or to correspondence searching in depth from stereo) and because repositioning the bulky acquisition devices between views is difficult. Registration is difficult and requires human assistance in the form of correspondences between features across views. Model construction is slow because the registered color and geometry dataset is huge. The lengthy modeling cycle limits the number of acquisition views.

A few views from different directions suffice for a good model in the outside-looking-in case where objects are viewed from outside their bounding volume. Examples are scanning a statuette on a rotating platter, scanning a piston for reverse engineering, or scanning an ancient throne from all sides. However, many views are needed in the inside-looking-out case where we wish to explore a scene from within. A few views cannot produce a good model even with careful view planning [Maver 1993, Allen 1998, Scott 2001]. We base this claim on extensive modeling experience with a laser rangefinder. Acquiring a room from ten views takes an entire day and model construction takes another day, yet incomplete models are obtained. Many more views are required to capture the missing data because it is scattered throughout the scene. Each view has the same high cost, but provides little new data.

We propose an interactive modeling paradigm in which an operator acquires thousands of views by scanning the scene with a portable acquisition device. The views are registered and are merged into an evolving model that is continually displayed for immediate operator feedback. The operator builds a complete model by checking the display for missing or undersampled regions and aiming the acquisition device at them. No special training or expensive equipment is required.

We have built a prototype interactive scene modeling system, called the ModelCamera, that processes five views per second. The acquisition device is a video camera with an attached laser system that provides 49 depth samples per video frame (Figure 1). The sparse depth sampling is dictated by the need for speed. We sample the scene densely by pooling the sparse samples from many frames. We register quickly by exploiting the close spacing between frames to simplify depth and color matching. Scene fiducials and trackers are avoided because they are impractical for large scenes. The close spacing between frames also makes it easy to construct the model incrementally, since each frame adds little new data.

We model scenes in two modes based on their geometric complexity. Structured scenes consist of large smooth surfaces, such as doors, walls, and furniture (Figure 2). Unstructured scenes consist of small uneven surfaces, such as a plant (Figure 3), a messy bookshelf, or coats on a rack. A structured scene contains

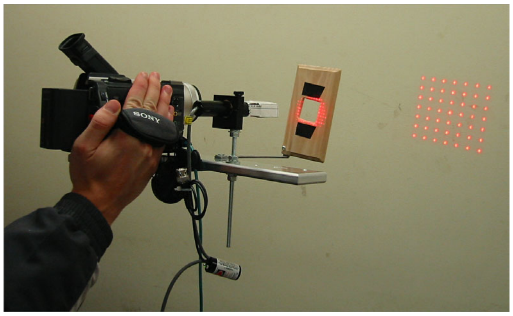


Figure 1: *ModelCamera.*

1-3 surfaces per frame, versus 10-100 in an unstructured scene. Our website http://www.cs.purdue.edu/cgvlab/modelCamera/modelCamera.html shows more models.

This paper surveys the ModelCamera project. The prototype and the structured scene mode were previously described in [Popescu 2003, 2004]. We have extended interactive modeling to unstructured scenes using Depth Enhanced Panoramas.

2. Prior work

We classify prior work in automated scene modeling by the depth acquisition methodology.

Modeling without depth

Some modeling techniques avoid depth acquisition altogether. QuickTime VR panoramas [Chen 1995] 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, which makes them popular in online advertisement. 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. Light fields [Levoy 1996, Gortler 1996] 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 and refraction. Light fields are constructed from a large set of registered photographs. Acquiring and registering the photographs is challenging. Another disadvantage is that the database is impractically large for complex scenes. Our approach addresses these problems.

User-specified 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 [Debevec 1996]. The geometric part of the hybrid geometry-image-based representation is created from user input in [Hubbold 2002]. In view morphing [Seitz 1996], the user specifies depth in the form of correspondences between reference images. Another example is image-based editing [Anjyo 1997, Oh 2001], which builds 3D models by segmenting images into sprites that are mapped to separate planes. User-specified 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 acquisition is slow and difficult.

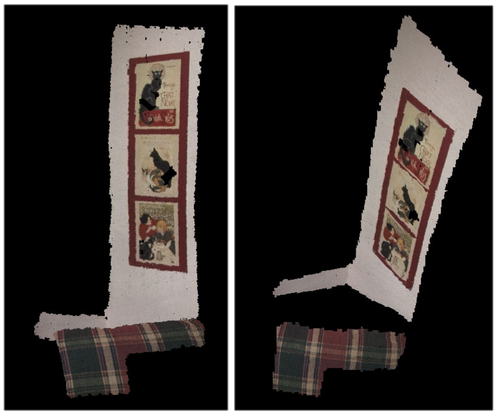


Figure 2: *Room fragment modeled freehand in 28s with 133 frames.*

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 [Levoy 2000, Bernardini 2002], of Jefferson's Monticello [Williams 2003], of cultural treasures of Ancient Egypt [Farouk 2003], of the Parthenon [Stumpfel 2003], and of the ancient city of Sagalassos [Pollefeys 2001, 2002]. The main disadvantage of this approach is the long per-view acquisition time, which limits the number of views. This in turn leads to incomplete models, especially in the inside-looking-out case where the device is surrounded by the scene. Another disadvantage is the high equipment cost.

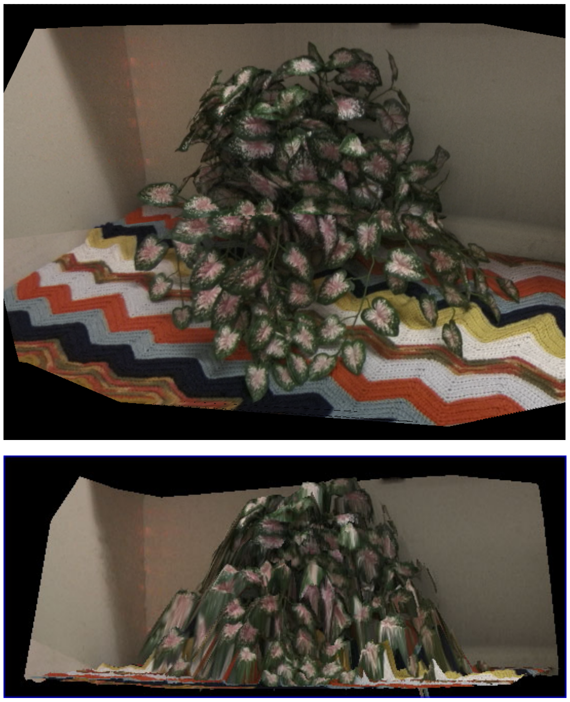


Figure 3: *Plant modeled in 120s from 500 frames.*

Interactive depth

If a small part of the scene is acquired at each view, the per-view depth acquisition task is simplified and can be carried out by portable devices. Several hand-held depth acquisition devices have recently been developed.

One architecture is a fixed camera and a mobile light-pattern source. One variant [Takatsuka 1999] uses a hand-held laser point projector on which three green LED's are mounted. The position of the LED's in the camera frame is used to infer the position and orientation of the laser beam. The red laser dot is detected in the frame and then triangulated as the intersection between the pixel ray and the laser beam. Another variant [Bouguet 1999] extracts depth from the shadow of a rod captured by a camera under calibrated lighting. Another architecture [Borghese 1998] uses two cameras mounted on a tripod and a hand-held laser point projector. The main problem with these systems is that they are limited to a single view by the fixed camera.

Hebert [2001] proposes a system where the operator can freely change the view. The device consists of two cameras and a cross-hair laser light projector. Frame to frame registration is achieved using a set of fixed points projected with an additional, fixed laser system. The fixed points are easy to discern from the cross-hair and act as fiducials. The system is not well suited for large scenes, since a large number of fiducials would be needed. It acquires depth only over a very narrow field of view at each frame, which implies long acquisition times in the case of complex scenes. It does not acquire color.

Rusinkiewicz et al. [2002] present an object modeling system based on structured light. The object is maneuvered in the fields of view of a fixed projector and camera. The frames are registered in real time using an iterative closest point algorithm. The evolving model is constructed in real time and is rendered to provide immediate feedback to the operator. The system does not acquire color. The modeling paradigm appears inapplicable to scenes. A similar system is proposed by Koninckx [2003] where moving or deformable objects are captured in real time. The system acquires depth using a pattern of equidistant black and white stripes and a few transversal color stripes for decoding. 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.

3. Acquisition device

The design criteria are real-time color-and-depth acquisition and freehand operation. We have developed an acquisition device that consists of a hand-held digital video camera enhanced with a laser system (Figure 1).

We use a high-end consumer-level digital video camera that weighs 1kg, has a CCD resolution of 720x480x3, costs $1,500, and operates in progressive scan mode at 15 fps. The laser system projects a pattern of 7x7 laser beams into the field of view of the video camera. It consists of a single laser source and a diffraction grating that acts as a beam splitter [Stockeryale]. It weighs less than 100g, costs $1,000, is eye safe (class IIIa), and is powerful enough to produce bright dots in the video frame when used indoors. It is rigidly attached to the camera with a custom 250g bracket that we designed to deflect less than 1mm under a 2kg force.

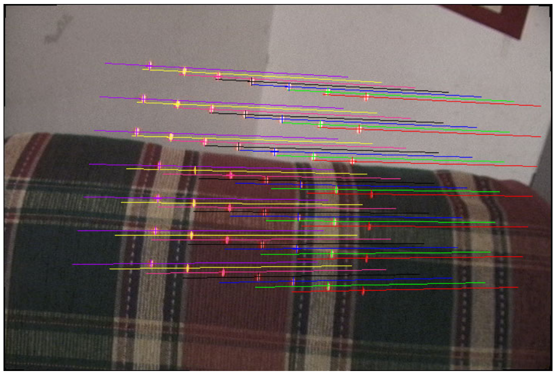


Figure 4: *Frame with 49 dots detected along epipolar segments.*

The video frames are read into a PC in real time through a FireWire interface. The frame is undistorted, the dots are located in the frame, and their 3D positions are computed by triangulation between the optical rays and the laser beams. Each dot is restricted to a fixed epipolar line segment (Figure 4) because the lasers are fixed with respect to the camera. The epipolar geometry constraint and frame to frame coherence make depth acquisition very efficient. The epipolar segments are disjoint to avoid dot confusion.

Calibration

We have developed a 5 minute calibration procedure for the ModelCamera that first calibrates the video camera [Bouguet www] (pinhole model with 5 distortion coefficients [OpenCV www], average calibration error 0.1 pixels), then finds the 2D epipolar lines (average 2D line fitting error 0.3 pixels), and finally finds each beam’s 3D equation in the camera coordinate system (average 3D line fitting error 1.5 mm).

Dot detection

The dot detector finds intensity peaks along the epipolar segments (Figure 5). Candidate peaks have to pass additional 2D symmetry tests. We exploit coherent camera motion by starting the search at the dot from the previous frame. This heuristic fails when a dot jumps from one surface to another, and its entire epipolar segment is then searched. Dot detection works well on our test structured scenes: 99% success at 70 cm and 85% at 200 cm. Unstructured scenes are harder because of laser scattering, reflection, and occlusion. False positives are minimized by requiring that a dot appear at roughly the same place in several frames before adding it to the model. We also narrow the range of potential depth values, which shortens the epipolar segments and further reduces false positives. In our unstructured scenes, 60% of the dots are detected.

Dot detection is extremely fast, taking less than 5ms per frame (all timing information reported in this paper is for a 2GHz 2GB Pentium Xeon PC). The *depth accuracy* is a function of the dot detection accuracy, of the camera field of view, of the frame resolution, and of the baseline. For a baseline of 15 cm, a one-pixel dot detection error translates into a depth error of 0.1 cm at 50 cm, 0.35 cm at 100 cm, 1.5 cm at 200 cm and 3.5 cm at 300 cm. We estimated dot detection accuracy by scanning a white wall from several distances and measuring the out-of-plane displacements of the triangulated 3D points. At 200 cm, the average/maximum displacements were 0.33 cm/1.1 cm, which indicates a dot detection error of 0.5 pixels. Better results were obtained at shorter distances.

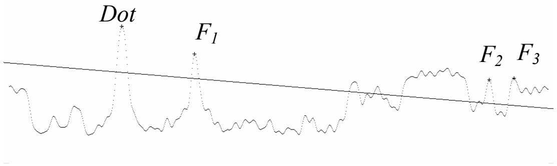


Figure 5: *Intensity along epipolar line with dot and false peaks. Line indicates threshold.*

Modeling power

A rough comparison of the ModelCamera with a typical laser rangefinder shows that sequences of sparse depth views have ample modeling power. The ModelCamera acquires 700,000 depth samples per hour (49 depth samples per frame x 80% dot detection success rate x 5 frames per second x 3,600 seconds). Counting two triangles per depth sample, the acquisition rate is 1,400,000 triangles per hour. Relying on the real-time feedback, the operator avoids oversampling low curvature surfaces and concentrates on the parts of the scene with higher geometric complexity. This ensures that most of the acquired samples are relevant and are used in the final geometric model. Even if the ModelCamera is active only 30 minutes per hour, and even if only half of the raw triangles make it in the final model, the net acquisition rate of 350,000 triangles per hour is far higher than with prior systems.

From our experience with acquiring room-sized environments using a laser scanner, acquisition requires at least one hour per view, including the time needed for view planning and repositioning of the device. View registration and model construction add at least two hours per view. Thus acquiring and processing 8 views of a room takes at least 24 hours. After removing the unnecessary depth samples on the flat surfaces, the resulting geometric model of a room typically comprises a few hundred thousand triangles. The net modeling rate is 10,000-20,000 triangles per hour. Moreover, many surfaces are missed by the limited number of views.

Results summary

We have designed and built a $3,000 acquisition device from off-the-shelf components, we have devised a fast, accurate calibration routine, and we have developed a real-time dot detection algorithm. The ModelCamera acquires high-quality 720x480 video frames enhanced with 49 depth samples at the rate of 15 fps and with errors below 1cm.

4. Structured scenes

The color and depth data are given in camera coordinates, which change as the camera moves. The data is registered in the initial camera coordinate system. The transformation from the current frame to the initial frame is obtained by composing the motions between consecutive frames.

The motion between two frames is computed in three stages: 1) identify the surfaces in each frame; 2) compute a motion that minimizes the distance between the new laser dots and the old surfaces; and 3) extend the motion to minimize the color difference between selected new rays and the corresponding points on the old surfaces. The depth error is a smooth function, so it can be minimized by least squares. The minimization determines the component of the motion that is perpendicular to the scene surfaces, which comprises 3 of the 6 camera degrees of freedom. The color error is sensitive to the other 3 degrees of freedom, which represent parallel motion. Iterative minimization is required because the color error is irregular. Depth registration allows for a fast, robust solution by reducing the search space dimension from 6 to 3.

Our algorithm improves upon the iterative closest point algorithm (ICP) [Besl 92], which is the state of the art in interactive registration [Rusinkiewicz 2002]. ICP registers two dense depth samples by iteratively forming correspondences between the samples and minimizing the depth error of the corresponding elements. The inner loop is essentially our depth registration algorithm. Hence, ICP cannot detect parallel motion or other motions along symmetry axes. We solve this problem with color registration. Moreover, we make do with sparse depth, which is easy to acquire and process interactively (49 dots versus thousands of depth samples).

4.1. Surface identification

The dots in a frame are grouped into surfaces. For example, the frame in Figure 4 contains three surfaces: the bottom four rows of dots lie on the couch backrest, the three right dots of the top three rows lie on the right wall, and the remaining dots lie on the left wall. Each row and column of dots is examined for surface boundaries. The boundary can be a depth discontinuity, such as where the visible part of the backrest ends and the walls appear, or a depth derivative discontinuity, such as where the walls meet.

A dot connectivity graph is constructed by linking every dot to its left, right, bottom, and top neighbors then breaking the links that span boundaries. A boundary is detected by thresholding the curvature. Using a depth first traversal, the graph is partitioned into connected components that represent surfaces. Cubic polynomials $z=p(x,y)$ are least-squares fitted to the surfaces.

4.2. Depth registration

We perform depth registration by formulating linearized depth equations and solving them by least squares. The depth equations state that the new dots lie on the surfaces of the corresponding old dots. Symmetric surfaces lead to non-generic equations that have multiple solutions. A surface is symmetric when it is invariant under translation along an axis, rotation around an axis, or coupled translation and rotation. Examples are planes, surfaces of extrusion, surfaces of rotation, and spheres. The distance from the dots to a symmetric surface is constant when the camera performs these motions. We restrict depth registration to 3 asymmetric motions that we identify using surface normals.

4.3. Color registration

We compute the other 3 motions by minimizing a color error function. The error of a pixel in the new frame is the RGB distance between its color and the color where it projects in the old frame. The old color is computed by bilinear interpolation. We minimize the sum of the pixel color errors by the downhill simplex method. This method is simple and does not require derivatives, which are expensive to compute. The pixels are assigned depths by linear interpolation from the three nearest dots. They are projected into the old frame by incremental 3D warping [McMillan 1995, McMillan 1997]. Warped-image reconstruction is unnecessary for error evaluation, so this approach does not incur the full cost of IBR by 3D warping [Popescu 2003].

4.4. Model construction

The scene is modeled as a collection of depth images that are created on demand as modeling progresses. We use depth images because they can be transformed and merged efficiently [Shade 1998, Popescu 2003].

The region spanned by the dots is triangulated. Each color pixel in the region is assigned a depth value from the triangulation. The color/depth samples are added to the model. When the new frame contributes a sample approximately at the same distance as a prior sample, the better sample is retained. The quality metric is based on the sampling rate of the current surface. The operator can select a visualization mode that highlights the parts of the model that were acquired below or above the desired sampling rate. Samples that are well behind or in front of a prior sample are added to a new image. Samples that project at the border between two depth images are repeated to provide overlap. The model quality is improved using subpixel offsets [Popescu 2004].

The depth images are transformed into texture-mapped triangle meshes that are rendered to provide operator feedback. Figure 6 shows the feedback provided to the operator: current frame (bottom left of the feedback window), 3D view of the evolving model, and depth image frusta (green “flies” around the surfaces); in the bottom image, the model depth images are shown in wireframe with different colors.

4.5. Results

We have tested the registration algorithm on thousands of frames in the room scene. Surface identification is accurate and robust based on manual verification and visual inspection of the resulting models. Every surface was found. No dot was assigned to an incorrect surface, although occasionally a dot that lay on a surface was unassigned. The average surface fitting error was 0.2cm and no frame was rejected because of a large error. Registration succeeded in 99% of the frames. When it failed, we found it easy to restore registration using the immediate graphical feedback. The average/maximum registration times were 100ms/200ms; 95% of the time was spent in color error evaluation. For our test scenes, the average/maximum model construction times were 60/120ms. Modeling and interactive visualization scale well and are robust. A depth image of 256 x 256 pixels and a triangulation step of 8 pixels, yields 2K triangles and 256 KB of texture. Current graphics hardware can easily handle 100 depth images.

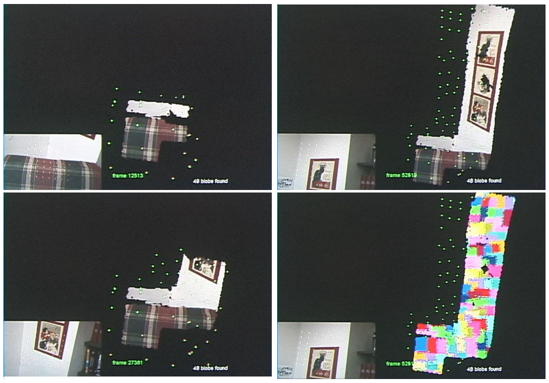


Figure 6: *Snapshots of the operator feedback window.*

5. Unstructured scenes: depth enhanced panoramas

An unstructured scene consists of many small surfaces. Each surface contains too few laser dots for an accurate polynomial fit and the depth-then-color registration algorithm fails. We model unstructured scenes by mounting the ModelCamera in a bracket that allows it to pan and tilt around the camera's center of projection (Figure 7). As the operator sweeps the scene, the ModelCamera acquires a sequence of dense color and sparse frames as before. The frames are registered using the color data only and are merged into an evolving scene model, called a depth enhanced panorama (DEP). The DEP is displayed continually to provide immediate feedback to the operator.

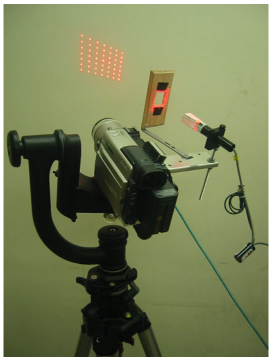


Figure 7: *ModelCamera mounted in parallax-free pan-tilt bracket.*

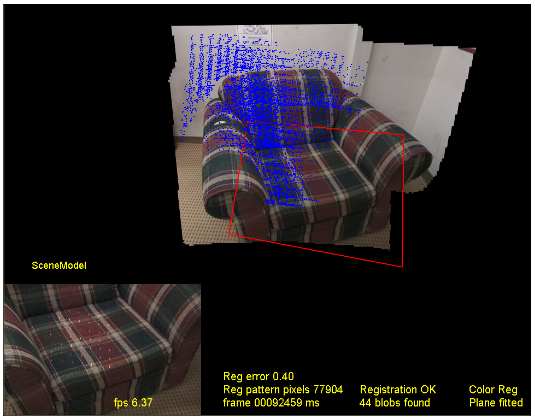
Besides providing a solution for difficult to model unstructured scenes, DEP's are a powerful method for modeling and rendering indoor scenes. DEP’s remove the fundamental limitation of color panoramas [Chen 1995] by supporting viewpoint translation, yet retain their speed, convenience, and low cost.

5.1. 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. The error of a pixel in the current frame is the RGB distance between its color and the color where it projects in the cube map. We select a registration pixel pattern in the current frame consisting of horizontal and vertical segments that exhibit considerable color variation. 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 and 2 divisions. We minimize the sum of the square of the pixel errors by the downhill simplex method.

The registered frames are merged into an evolving DEP. The color data is merged into a cube map panorama. The faces of the cube map are divided into tiles. For efficiency, the current frame updates only the tiles that fall within its field of view and are not yet complete. Registration takes 150ms per frame and merging takes 50ms per frame, so the modeling rate is 5 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 (Figure 8).



fps 6.37

Reg error 0.40
Reg pattern pixels 77904
frame 00092459 ms

Registration OK
44 blobs found

Color Reg
Plane fitted

Figure 8: *Operator feedback: current frame (*bottom left*), last registered frame (*red rectangle*), depth samples (*blue*).*

The video camera adjusts the white balance automatically as darker/brighter parts of the scene are scanned, which assigns the same diffuse surface different colors in different frames. The dynamic range problem is milder than in outdoor scenes. If the operator moves slowly between dark and bright regions, registration is robust. New samples are blended with the old samples to obtain a better texture uniformity (Figure 9). Blending also hides red artifacts due to laser scattering on thin or shiny surfaces.

5.2. DEP visualization

We have developed a DEP visualization method that produces high-quality images of the scene at interactive rates. The method supports real-time visualization of evolving DEP's, which is integral to interactive modeling. We triangulate 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. The 3D triangle mesh is texture-mapped with the cube map faces.

During 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 [Devillers 1992a and 1992b, and Boissonnat 1993]. The implementation was obtained from [Delaunay www]. The DEP's in Figure 11 contain 17—55 thousand triangles and were acquired in between 1—5 minutes. DEP's capture the appearance of such complex objects well.

6. Conclusions and future work

We have presented an interactive scene modeling system based on dense color and sparse depth. The operator scans structured scenes freehand with a portable acquisition device. Unstructured scenes are modeled using a parallax-free pan-tilt bracket. The system acquires video frames, extracts depth samples, registers the frames, and merges them into an evolving model that is rendered continually for operator feedback. This pipeline runs at five frames per second.

Our research shows that sparse depth (and dense color) has the power to model complex scenes. Acquiring only 49 depth samples per frame is compensated for by the fast pipeline. Although each frame is registered accurately with respect to the previous frame, small registration errors can accumulate over long frame sequences. We plan to eliminate drift using scene features as fiducials.

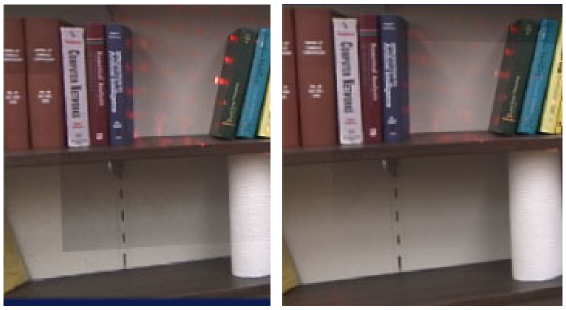


Figure 9: *Face fragment without (*left*) and with blending (*right*).*

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. We will continue to develop DEP's. 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 union their individual geometries. The challenge here is to combine two approximate representations into a better representation.

Another research path is to use the texture information to improve the geometry. Presently, accurate geometry edges can only be obtained if the operator overscans the edge region to ensure that sufficient depth samples lie on the edge. Edges could be detected automatically in the texture and used to interpolate additional depth samples on the edge.

We are designing a new ModelCamera prototype with a custom laser system that is brighter and acquires 100-200 depth samples per frame, which will bring us closer to our goal of modeling one room in one hour and entire buildings in a single day by scanning in parallel.

7. Acknowledgments

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, by a Purdue University OTC Trask award, and by the Purdue University Center for Graphics, Geometry, and Visualization.

References

[Allen 1998] P. Allen, M. Reed, and I. Stamos, View Planning for Site Modeling Proc. DARPA Image Understanding Workshop, November 21-23, 1998

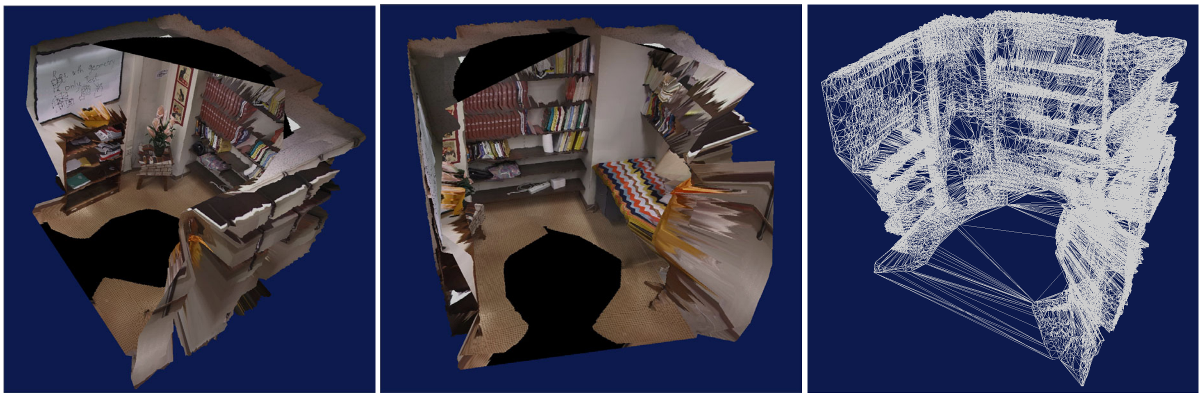


Figure 10*: Depth enhanced panorama of a room: 100,000 triangles acquired in 30 minutes.*

[Anjyo 1997] Anjyo, K., Horry, Y., and Arai, K. "Tour into the Picture" Proc. SIGGRAPH '97 pp. 225-232.

[Bernardini 2002] 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.

[Besl 1992] P. Besl, N. McKay. A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239-256.

[Borghese 1998] N. A. Borghese et al., Autoscan: A Flexible and Portable 3D Scanner, IEEE Computer Graphics and Applications, Vol.18, No.3, 1998, pp. 38-41.

[Bouguet 1999] J.-Y. Bouguet and P. Perona, 3D Photography using Shadows in Dual-Space Geometry, International Journal of Computer Vision, Vol. 35, No. 2, 1999, pp. 129-149.

[Bouguet www] http://www.vision.caltech.edu/bouguetj/calib_doc

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

[Debevec 1996] 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).

[Farouk 2003] 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.

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

[Hebert 2001] P. Hebert, A self-referenced hand-held range sensor, Proceedings of Third International Conference on 3-D Digital Imaging and Modeling, pp. 5-12, 2001.

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

[Koninckx 2003] 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.

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

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

[Maver 1993] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view, IEEE Transactions on Pattern Analysis and Machine Intelligence 15(5), pp. 417-433, 1993.

[McMillan 1995] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In Proc. SIGGRAPH '95, pages 39-46, 1995.

[McMillan 1997] L. McMillan. An image-based approach to three dimensional computer graphics. Ph.d., University of North Carolina at Chapel Hill, 1997.

[Oh 2001] Byong Mok Oh, Max Chen, Julie Dorsey, and Fredo

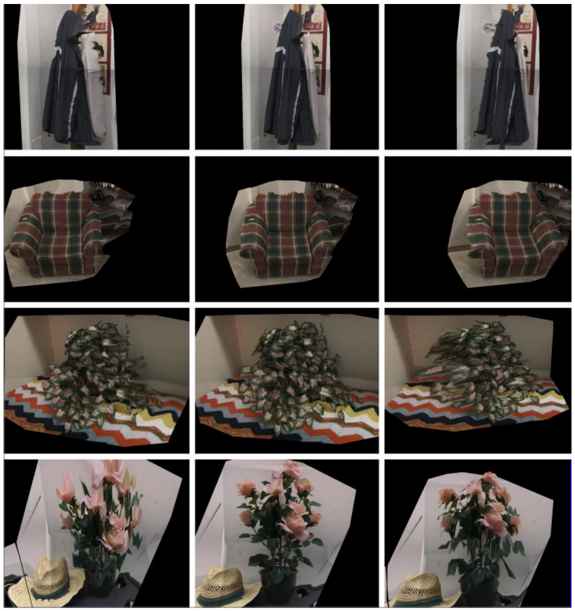


Figure 11*: Depth enhanced panoramas. Images in center column were rendered from the panorama center. The left and right columns show views translated away from*

Durand. Image-Based Modeling and Photo-Editing Proceedings SIGGRAPH 2001.

[OpenCV www] http://www.intel.com/research/mrl/research/opencv/index.htm

[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).

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

[Pollefeys 2001] M. Pollefeys, L. Van Gool, I. Akkermans, D. De Becker, "A Guided Tour to Virtual Sagalassos", Proc. VAST2001 (Virtual Reality, Archaeology, and Cultural Heritage)

[Popescu 99] Popescu V., and Lastra A., "High Quality 3D Image Warping by Separating Visibility from Reconstruction", *UNC Computer Science Technical Report TR99-017*, University of North Carolina, (1999).

[Popescu 2000] Voicu Popescu et al. The WarpEngine: An architecture for the post-polygonal age. Proc. ACM SIGGRAPH, 2000.

[Popescu 2003] 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.

[Popescu 2004] V. Popescu, E. Sacks, and G. Bahmutov. Interactive Point-Based Modeling from Dense Color and Sparse Depth. Proc. of Eurographics Symposium on Point-Based Graphics, Zurich, 2004.

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

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

[Scott 2001] W. Scott et al., View Planning with a Registration Constraint, In IEEE Third International Conference on 3D Digital Imaging and Modeling, Quebec City, Canada, May 28 – June 1, 2001.

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

[Shade 1998] Jonathan Shade et al. Layered Depth Images, In Proceedings of SIGGRAPH 98, 231-242.

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

[Stumpfel 2003] 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.

[Takatsuka 1999] M. Takatsuka et al., Low-cost Interactive Active Monocular Range Finder, in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 1999, pp. 444-449.

[Williams 2003] 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).