Eurographics Symposium on Point-Based Graphics (2004)
M. Alexa, S. Rusinkiewicz, (Editors)

Interactive Point-Based Modeling from Dense Color and Sparse Depth

Voicu Popescu, Elisha Sacks, and Gleb Bahmutov

Computer Science Department, Purdue University, USA

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. This paper discusses interactive modeling of structured scenes, which consist of large smooth surfaces. We have built an acquisition device that captures 7x7 evenly spaced depth samples per frame. The samples are grouped into patches that are approximated with polynomial surfaces. Consecutive frames are registered by computing a motion that aligns their depth and color samples. The scene is modeled as a collection of depth images created on demand during scanning. Resampling errors are avoided by using offsets to record accurately the positions of the acquired samples. The interactive modeling pipeline runs at five frames per second.

Categories and Subject Descriptors (according to ACM CCS): I.3.3. [Computer Graphics]—Three-Dimensional Graphics and Realism.

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

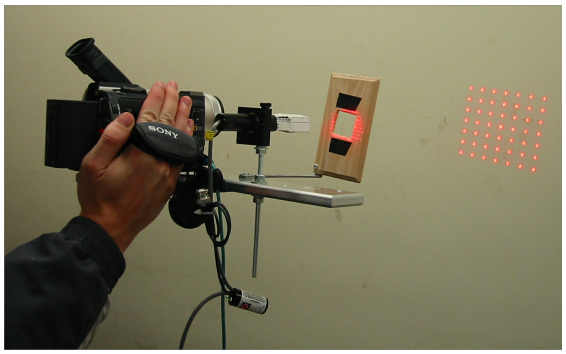

Figure 1: *Prototype acquisition device.*

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 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.

This paper discusses structured scene modeling. Structured scenes consist of large smooth surfaces, such as doors, walls, and furniture. They are acquired freehand for maximum maneuverability (Figure 2 and the video submission). Unstructured scenes consist of small uneven surfaces, such as a plant, a messy bookshelf, or coats on a rack. They are discussed in the final section.

2. Prior work

Modeling without depth

Some modeling techniques avoid depth acquisition altogether. QuickTime VR panoramas [Chen 1995] are 2D

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

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.

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

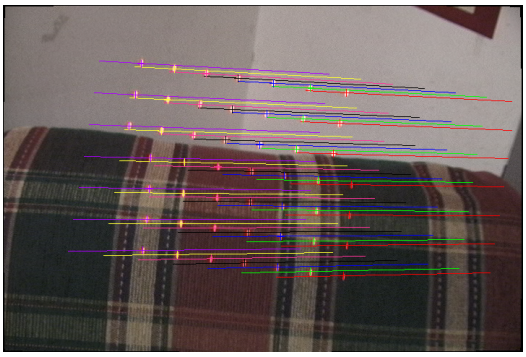

Figure 3: *Frame with 49 dots detected along epipolar segments.*

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.

Interactive depth

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

Our device (Figure 1) consists of a video camera and a laser system. The camera weighs 1kg, has a CCD resolution of 720x480x3, costs $1,500, and operates in progressive scan mode at 15fps. The laser system consists of a laser and a beam splitter that generates a 7x7 square pattern [Stockeryale]. It weighs 100g, costs $1,000, is eye safe (class IIIa), and produces bright dots in indoor scenes. The laser system is rigidly attached to the camera with a custom 250g bracket that we designed to deflect less than 1mm under a 2kg force. The camera is connected to a PC (2GHz 2GB Pentium Xeon) by a FireWire interface.

Depth samples are obtained by undistorting the frame, finding its laser dots, and computing their 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 segments disjoint, which prevents ambiguity in dot/laser assignment (Figure 3). We use a dot detection algorithm similar to the one described in [Popescu 2003].

The system acquires 720x480 video frames enhanced with 49 evenly spaced depth samples. The acquisition rate is 15 frames per second. 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 rate is 99% on smooth surfaces at 70cm, 85% at 200cm, and 60% on unstructured scenes. 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.

4. Registration

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 3 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

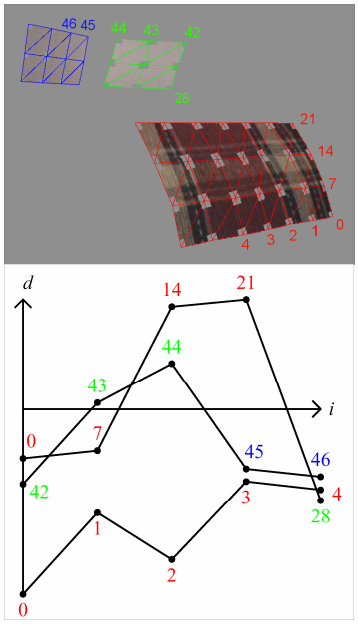

Figure 4: *Surface identification for Figure 3 frame.*

dots, called a strip, 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. Given a strip of n dots with depths $z_1, \ldots, z_n$, we compute the second differences $d_i = z_{i+2} - 2z_{i+1} + z_i$ to approximate the curvature along the strip. A depth derivative discontinuity occurs between dots j and $j+1$ when d_j and d_{j+1} are large, and a depth discontinuity occurs when they are very large. A threshold of 3 is used for boundary detection.

Figure 4 plots $log(d/3)$ against i for the bottom row, the top row, and the right column of Figure 3, using the same dot numbers and colors. Strips are broken at peaks that cross the horizontal axis. The bottom row lies well below the axis, the top row has a large peak at dot 44 where the walls meet, and the right column has a very large peak at dot 21 where the backrest ends.

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. 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. The dots are mapped to surface points by perpendicular projection. The frame is rejected if the mean dot/point distance exceeds twice the dot detection accuracy. Otherwise, the dots are assigned the surface normals of their surface points.

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. An equation is formulated for an old/new dot pair when both dots have four surface neighbors, which indicates that they are interior to the surface. Dots on surface boundaries are skipped because their normals can be inaccurate.

The old surface is linearized as $n(p-a) = 0$ with n the surface normal, p the new dot, and a the old dot. The motion is $m(p) = t + Rp$ with t a translation vector and R the matrix that rotates around axis d by angle θ. The motion is linearized as $m(p) = t + p + r \times p$ with $r = \theta d$, and then is substituted into the linearized surface equation to obtain the depth equation $tn+r(p \times n) = n(a-p)$. The k depth equations form a system $Ax = b$ with A a k-by-6 matrix, $x = (t_x, t_y, t_z, r_x, r_y, r_z)$ a 6 vector, and b a k vector. The six elements of x represent the translations and rotations of the camera around the three coordinate axes.

A least-squares solution is an x that minimizes the geometric mean distance from the transformed dots to the surfaces. A generic system has a unique solution when $k \geq 6$, which holds in structured scenes. But 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 the depth equations to a 3-dimensional subspace of x that represents asymmetric motion. Any normal vector to a surface generates three asymmetric motions: translation along it and rotation around two perpendicular axes. A proof is obtained by checking the finite list of symmetric surfaces. We compute the normal at the centroid of the old dots and formulate the depth equations in a coordinate system where this normal is the z axis. Thus, x_1, x_2, and x_5 are possibly symmetric, while x_2, x_3, x_4, are always asymmetric. We drop the symmetric x_i's from the depth equations and solve for the others by singular value decomposition.

4.3. Color registration

We compute the symmetric x_i's 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 because the pixel projects at fractional coordinates. Small camera motions produce rapid, erratic changes in color error. We reduce the variability by convolving each frame with an 11-by-11 box filter. We then select a set of new pixels and minimize the sum of the squares of their errors by the downhill simplex method. This method is simple and does not require derivatives, which are expensive to compute.

The pixels are selected by scanning every kth row and column (we used $k = 20$) of the image and splitting them into segments. A segment is a maximal sequence of pixels that are dot free and that lie on a single surface. Dot pixels are excluded because their color comes from the lasers,

rather than from the scene. 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. 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.

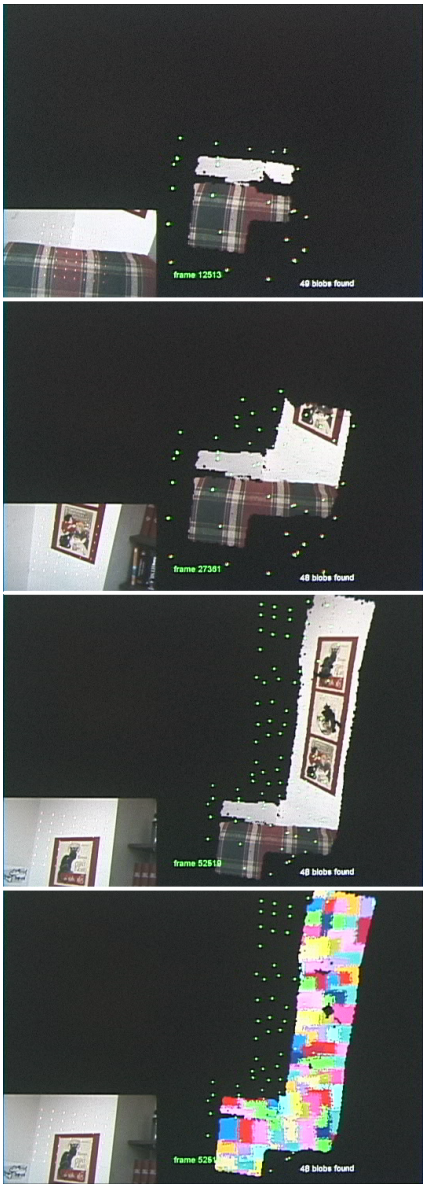

Figure 5: *Snapshots of the operator feedback window.*

5. Point-based 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]. Each registered frame is processed as follows. The region spanned by the dots is triangulated. Each color pixel in the region is assigned a depth value from the triangulation. The pixels that are illuminated by the lasers are excluded. Figure 4 shows the triangles with the excluded regions.

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 depth images are transformed into texture-mapped triangle meshes that are rendered to provide operator feedback. The mesh does not cross between surfaces separated by a depth discontinuity. We detect depth discontinuities in the model depth images by thresholding the local mesh curvature [Popescu 2000]. Figure 5 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.

5.1. Accurate modeling and rendering using offsets

The current-frame depth image is merged into the model by warping it to the relevant model depth images. Simply warping the samples is not sufficient: the model depth image needs to be reconstructed from the forward mapped samples to avoid holes. The reconstruction problem has received considerable attention from point-based modeling and rendering researchers.

McMillan [1997] introduces reconstruction by splatting, which approximates the footprint of the warped samples with simple image-space primitives (squares, rectangles, circles, ellipses). Popescu [1998, 2000] shows that a high-quality reconstruction of the warped image can be obtained by separating visibility from reconstruction. During the visibility stage, the position of the forward-mapped samples is recorded precisely using offsets. The offsets are used during the reconstruction stage to obtain a high-quality image. The *surfels* approach represents the scene with

small 3D surface primitives [Pfister 2000]. Surfels adopt the idea of separating visibility from reconstruction; *visibility splatting* is used to ensure that the final image is reconstructed only from accurately placed, visible samples. The *QSplatting* approach stores the point-based model in a hierarchy of bounding spheres, which provides visibility-culling and level-of-detail adaptation [Rusinkiewicz 2000]. QSplats have the advantage of progressive refinement, an important feature in the case of massive models.

In our case, the model depth images are an intermediate representation that is used to render the scene from novel views. We have developed a depth image modeling and rendering technique that avoids resampling errors using offsets. Our method is related to the offset reconstruction and surfel techniques and has three stages (Figure 6).

Splatting The frame depth and color samples are splatted in the model depth image. For efficiency, we use square splats. The size of the splats is estimated conservatively to resolve visibility correctly. A pair of 4 bit offsets is used at each model depth image pixel to record the position of the warped sample within $1/8^{th}$ of a pixel. We store the 8 bits of offset in the alpha channel of the texture of the model depth image.

Depth image cleanup After splatting, the offsets encode the location of the acquired samples relative to the center of the pixel. The offsets are used to eliminate the imprecise

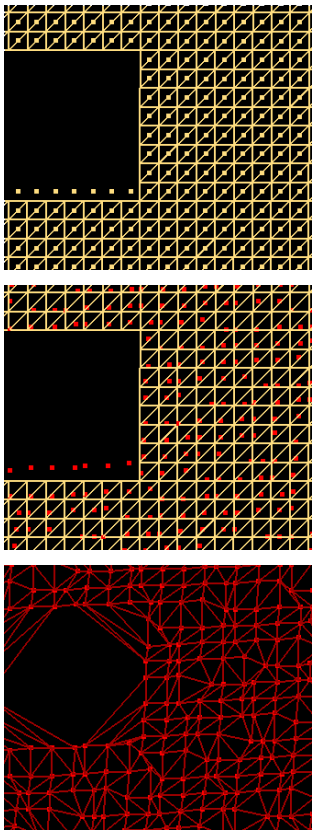

Figure 6: *Depth image fragment. Pixel grid with pixel centers (top), recovered original samples (middle) and Delaunay-triangulated original samples (bottom).*

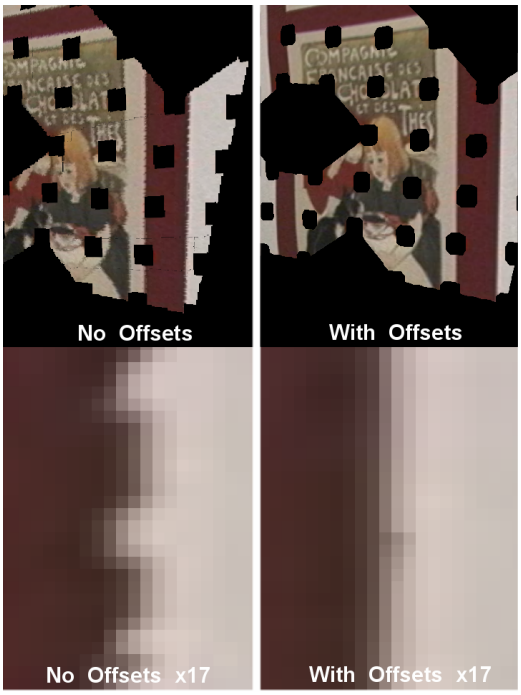

Figure 7: *Model depth image rendered from novel view with and without offsets.*

samples introduced by splatting and to recover precise, original samples, as acquired by the frame(s) that combined to fill in the model depth image. The original samples are gathered in a single pass over the model depth image from the pixels with offsets pointing within the current pixel.

Rendering The cleaned up depth image has occasional holes in the depth and color maps. The remaining, accurate samples are triangulated on the depth image plane. The triangulation is applied to the 3D original samples and the resulting 3D triangle mesh is rendered with per-vertex color in hardware.

Offsets improve the quality of the rendered images considerably, as seen in Figure 7. Offsets essentially correct the 3D position of the model samples. If no offsets are used, the color samples can move up to half a pixel in the construction depth image. This translates to large desired image errors when the desired view samples the scene surface more densely than the depth image or at a different angle. The method is efficient and can be applied after each frame to all depth images affected by the current frame.

6. Conclusions

We have presented an interactive scene modeling system based on dense color and sparse depth. The operator scans a structured scene freehand with a portable acquisition device. 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.

An earlier version of the ModelCamera system is described in [Popescu 2003]. That system used 16 separate

laser pointers, and was able to acquire only a single surface. The system described here introduces an improved acquisition device, a surface identification algorithm, multiple surface registration, and an accurate modeling algorithm based on offsets.

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. In one minute of operation, our system acquires about 12,000 depth samples. The operator aims the device at the parts of the scene with higher geometric complexity, thus most of the depth samples are relevant.

Point-based modeling and rendering scales well and is robust (Figure 8). 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.

The depth-then-color registration algorithm fails on unstructured scenes because it cannot identify any surfaces. We are developing an interactive modeling technique for unstructured scenes that uses a tripod to limit the camera motion to panning and tilting about its center of projection. Sequences of same-center-of-projection frames can be registered using the color data only, the same way images are stitched to form color panoramas. We also acquire depth, which is used to support view point translation, thus eliminating the fundamental limitation of color panoramas.

7. Acknowledgments

We thank 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.

References

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

[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.

[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).

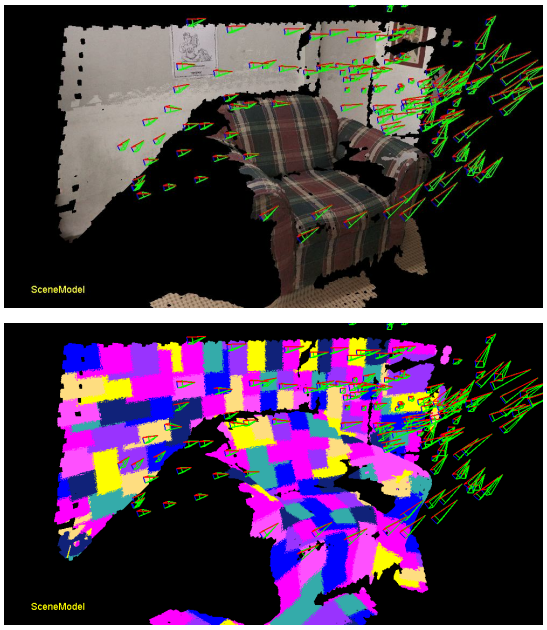

Figure 8: *Model obtained from a pre-registered sequence of 2300 frames.*

[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.

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

[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.

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

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

[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.

[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).