2004

Interactive Modeling from Dense Color and Sparse Depth

Voicu Popescu
Purdue University, popescu@cs.purdue.edu

Elisha Sacks
Purdue University, eps@cs.purdue.edu

Gleb Bahmotov

Report Number:

04-014

INTERACTIVE MODELING FROM DENSE COLOR AND SPARSE DEPTH

Voicu Popescu
Elisha Sacks
Gleb Bahmutov

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR #04-014
April 2004

Interactive Modeling from Dense Color and Sparse Depth

Voicu Popescu, Elisha Sacks, and Gleb Bahmutov
Computer Science, Purdue University

Abstract

Many computer graphics applications benefit from interactive rendering of real-world scenes. The challenge is to build a scene model that is sufficiently detailed to impart visual realism to the rendered images *and* is sufficiently compact to support interactive rendering. A promising approach is to build the scene model from acquired color and depth data. Systems that build high-quality models are slow (e.g. depth from stereo, laser rangefinding), require expensive and bulky equipment (e.g. laser rangefinding), and require manual intervention (e.g. image-based editing, view morphing, tour into the picture). Systems that do not suffer from these disadvantages build simplified scene models (e.g. color panoramas) that must be viewed from preferred locations.

We present a fast, easy to use, and inexpensive modeling system that builds scene models that support realistic interactive rendering from a wide range of viewing locations. Our system acquires video frames augmented with sparse depth samples. The frames are registered and merged into an evolving model at the rate of five frames per second. The model is displayed continually to provide immediate feedback to the operator.

1. Introduction

We present research in scene modeling. The task is to build digital models of complex scenes that support interactive, photorealistic rendering. Scene modeling is the bottleneck in computer graphics applications ranging from cultural preservation to forensics to marketing to gaming. Manual modeling is unacceptably labor intensive. Automated modeling, where color and depth data is measured using an acquisition device, is a natural alternative. Many automated modeling techniques have been developed, but none offer a complete, fast, and inexpensive solution to real-world scene modeling. The main challenge is depth acquisition.

1.1 Prior work in automated scene modeling

Acquired dense depth In order to capture the full 3D complexity of a real-world scene, one has to establish the 3D position of each surface sample. Depth from stereo, and structured-light 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].

An important disadvantage common to all modeling systems that acquire dense depth 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.

User-specified course depth Another solution to the depth acquisition problem is manual geometry data entry. An example is the Façade 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 coarse depth systems take advantage of the user's knowledge of the scene, which allows him to maximize the 3D effect while minimizing the amount of depth data. The disadvantage of the approach is the need for manual input.

No 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

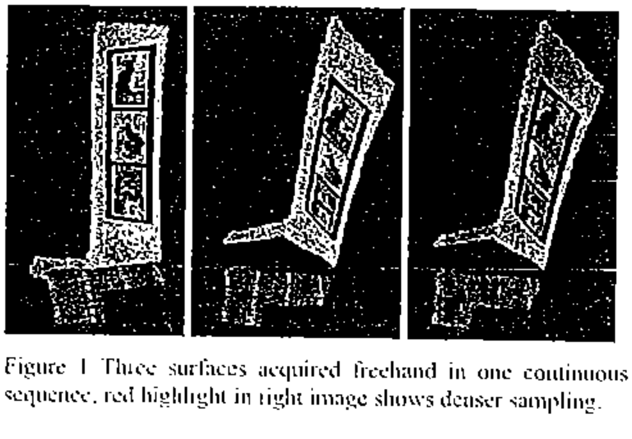

Figure 1 Three surfaces acquired freehand in one continuous sequence, red highlight in right image shows denser sampling.

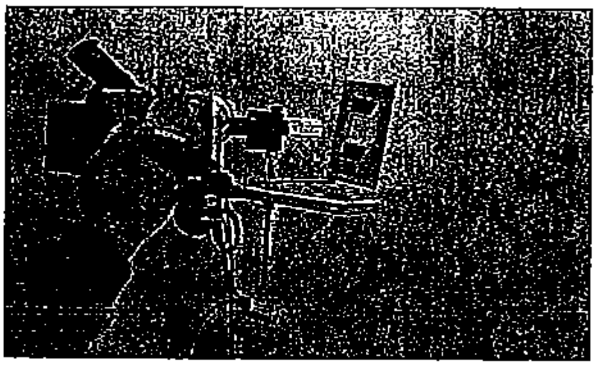

Figure 2 Acquisition device.

rapid, inexpensive acquisition and of interactive photorealistic 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.

Interactive modeling 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 is limited to the outside-looking-in modeling case and does not acquire color. 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.

1.2 Our approach

We present an interactive scene modeling system based upon the following requirements.

- *Fast modeling* The system should model a room in one hour.
- *Immediate feedback* The system should build the model incrementally and display it continually, so that the operator can validate the model right away.
- *Color acquisition* The models are intended for rendering, so high-quality color is needed.

Our system acquires dense color and sparse depth data. The data is registered and merged into an evolving model at the rate of five frames per second. The model is displayed continually to provide immediate feedback. We demonstrate the system on an office scene; we obtain meshes of 50,000 triangles in five minutes.

2. Acquisition device

Our acquisition device (Figure 2) consists of a digital video camera enhanced with a laser system that casts a pattern of 7x7 beams in its field of view. The pattern is created by splitting a main beam with a diffraction grating [Stockeryale]. The power of the laser system places it at the low end of class III-a, which is the classification of common laser pointers, and is thus eye safe. A bright dot is visible in the video frame where a beam intersects a scene surface. The dot is located in the video frame and its 3D position is triangulated by computing the intersection between the video camera ray and the precalibrated laser beam. We obtain 3 mm depth accuracy at 1m.

Our design provides an even depth sampling of the video frame. 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 is fast because the dots are confined to fixed epipolar segments in the video frames (Figure

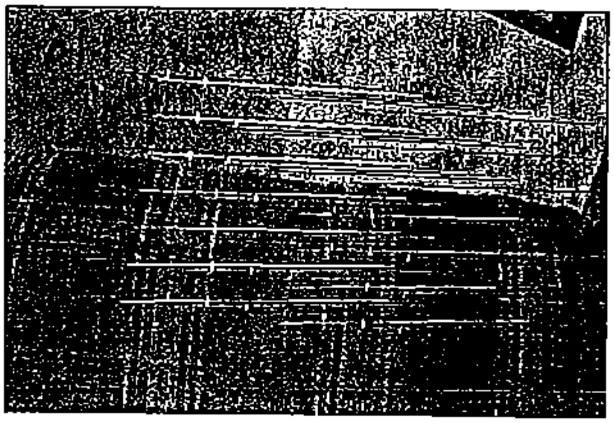

Figure 3 Frame with 49 dots detected along epipolar segments.

3). The lasers are configured to make the epipolar segments disjoint, which prevents dot detection ambiguity.

3. Structured scenes

We model scenes in two modes based on their geometric complexity. A scene is called structured when it consists of large smooth surfaces. The laser dots in each frame fall on a few surfaces. These surface patches can be modeled to the required accuracy by fitting polynomials through the dots. For example, Figure 3 shows one frame from a structured scene comprised of a couch, walls, and other furniture.

Structured scenes are modeled freehand. The operator sweeps the scene with the camera and the system acquires depth and color, registers this data in the coordinate system of the first frame, and merges it into an evolving scene model. We compute transformations between consecutive pairs of frames and compose them to obtain the transformation from each frame to the first frame. The transformation between an old/new frame pair equals the inter-frame camera motion. This motion is computed in three stages: 1) identify the surfaces in each frame; 2) compute a motion that minimizes the distance between the new 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.

3.1 Surface identification

The dots in a frame are grouped into surfaces. For example, 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 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.

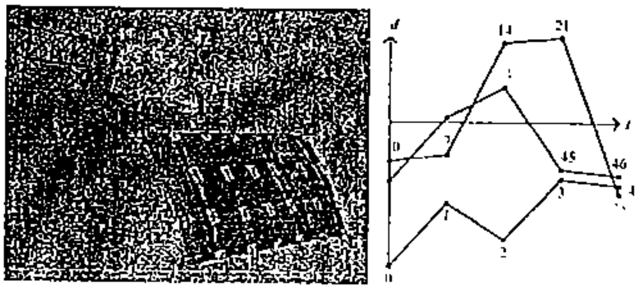

Figure 4 Surface identification for frame shown in Figure 3.

3.2 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, so adding them to a solution yields another solution.

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_6 are possibly symmetric, while x_3, x_4, x_5, are always asymmetric. We drop the symmetric x_i's from the depth equations and solve for the others by singular value decomposition.

3.3 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 *k*th 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].

3.4 *Modeling*

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 1996, 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 (samples per centimeter). 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 operator can select a visualization mode that highlights the parts of the model that were acquired below or above the desired sampling rate.

3.5 *Discussion*

The depth-then-color algorithm exploits the complementary properties of the depth and color errors. Depth error arises from motion perpendicular to a surface, while color registration arises from parallel motion. The depth error is a smooth analytic function, so it can be minimized by least squares, whereas the color error is jagged, so iterative minimization is required. Depth registration makes color registration fast and robust by reducing the search space dimension from 6 to 3.

Our algorithm improves upon the iterative closest point algorithm (ICP), 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 motion 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. Unstructured scenes

An unstructured scene consists of many small surfaces. Each surface contains too few laser dots for an accurate polynomial fit. Figure 9 shows typical unstructured scenes: a plant, books on a shelf, and coats on a rack. The depth-then-color registration algorithm fails on unstructured scenes because it cannot identify any surfaces.

We place the camera in a bracket that restricts its motion to panning and tilting around its center of projection (Figure 5). The operator scans the scene by panning and tilting the camera, color and depth are acquired as before, the two camera angles are computed from color, and an incremental model is built from the registered data. The model, called a depth-enhanced panorama, is an effective representation for unstructured scenes, which are difficult to model. Unlike regular panoramas, depth enhanced panoramas provide motion parallax when the view is translated (Figure 8, Figure 9).

Before modeling, the bracket is calibrated to make the rotation axes pass through the center of projection. The optimal setting is indicated by the absence of motion parallax between near and far objects when the camera is rotated. The bracket pan axis has to be expressed in the initial camera coordinate system. This calibration is performed transparently. The only requirement is that the camera be panned for at least 15 degrees before it is tilted.

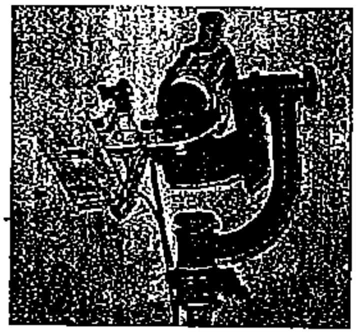

Figure 5 Parallax-free pan-tilt bracket.

The frames are registered the same way images are stitched together to form panoramas. We have developed a real-time stitching algorithm that registers the current frame with respect to the previous frame using a pattern of horizontal and vertical segments in the current frame. The algorithm finds the two camera angles by minimizing a color error function. The color error is computed by projecting every pixel of every segment of the registration pattern onto the previous frame and summing the squared color differences. The color at the projection location is computed by bilinear interpolation. The pixels are projected by projective texture mapping, which is less expensive than 3D warping. Each segment is projected incrementally at an average cost of 3 additions and 2 divisions.

Unstructured scenes have great color variation, so the global minimum valley is deep (Figure 7). The color error is minimized by downhill simplex, which quickly finds the bottom of the valley when started inside it. We predict the start point by assuming constant angular velocity and extrapolating from the previous frame. We sample the error in a rectangle around this point to detect rapid acceleration. Registration is fast and robust. In our experiments, we register hundreds of frames before a registration failure occurs. The system notifies the operator who returns to the latest registered frame and resumes modeling from there.

The registered frames are merged into a cube map panorama. The faces of the cube map are subdivided into tiles. The current frame

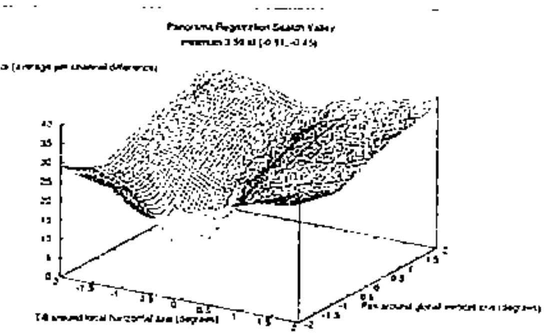

Figure 7 Color error as a function of pan and tilt angles.

updates only the tiles that fall within its field of view *and* are not yet complete. We allow the camera to adjust its settings dynamically to avoid under or over exposure. Registration is unaffected, except for increased error values at the adjusted frames. The adjustment causes color inconsistency in the merged panorama, which is visible in some of the panoramas shown in Figure 9. This prevents us from registering the new frame directly against the faces of the cube map, as would be preferable to avoid drift. Nevertheless, the drift is not significant: 3 degrees in an experiment where we panned the camera 360 degrees.

The system acquires depth at every registered frame. The operator can obtain a triangulation of the acquired dots in one second (Figure 6). The mesh topology is used to generate 3D triangles that can be rendered from any viewpoint.

Unstructured scenes are harder for dot detection because of laser scattering, reflection, and occlusion. False positives are minimized by requiring that a dot appear at roughly the same place in k = 3 frames before adding it to the model. We also narrow the range of potential z values, which shortens the epipolar segments and further reduces false positives. In our unstructured scenes, 60% of the dots are detected.

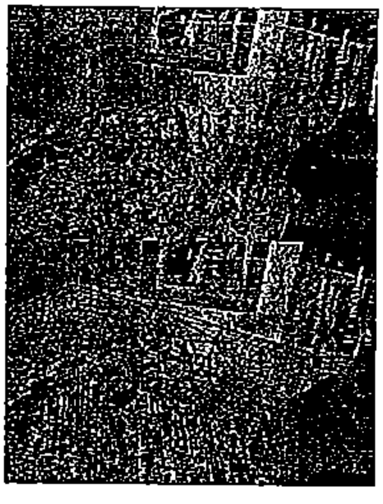

Figure 6 The dots are first triangulated in 2D on the face of the panorama (*orange*); then the connectivity inferred is used to make the 3D mesh (*blue*).

5. Results

Figure 1 shows three surfaces acquired freehand in 30s. Figure 9 shows 5 examples of depth enhanced panoramas acquired with our system, which contain between 17 and 55 thousand triangles and were acquired in between 1½ and 5 minutes. We refer the reader to the accompanying video for a good illustration of the motion parallax supported by depth enhanced panoramas.

6. Conclusions

We have presented an interactive scene modeling system based on dense color and sparse depth. The operator scans the scene with a portable acquisition device. Structured scenes can be scanned freehand, but the device must be mounted in a pan-tilt bracket to scan unstructured scenes. 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. We have demonstrated that interactive modeling is robust. We have acquired several sequences of thousands of frames without losing registration.

For modeling unstructured scenes we have introduced depth enhanced panoramas which keep the advantages of regular color panoramas (fast, inexpensive acquisition and high-quality interactive rendering), and remove the major limitation of regular

Figure 8 Depth enhanced panorama viewed from different view.

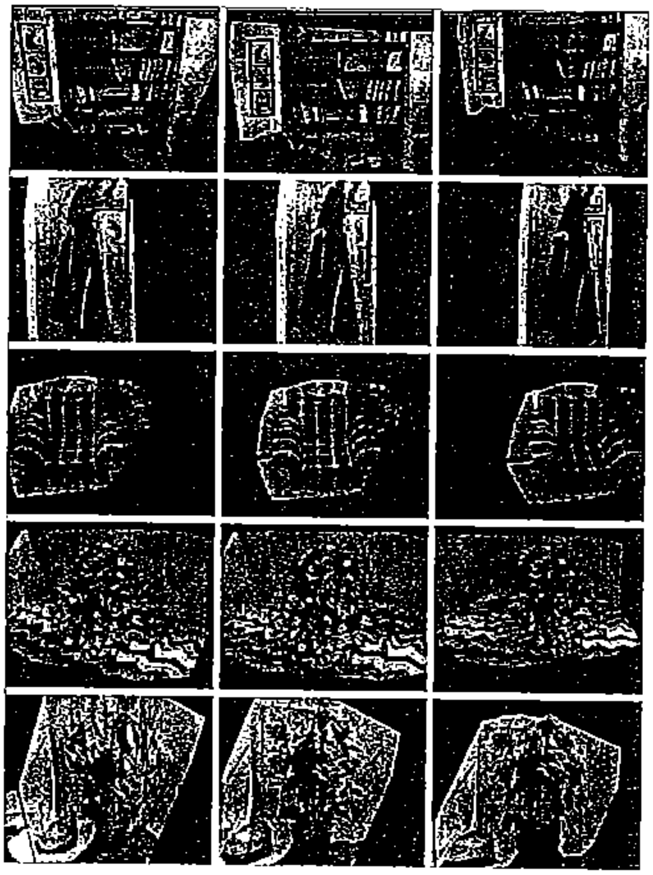

Figure 9 Depth enhanced panoramas. Images in center column were rendered from the panorama center. The left and right columns show views translated away from center.

color panoramas by supporting view translation. We will continue to develop depth enhanced panoramas. One possible path of future work is offline refinement. Edge detection and edge-aware triangulation could bring considerable improvement.

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. This indicates that sparse depth has the power to model complex scenes.

Although each frame is registered accurately with respect to the previous frame, small registration errors can accumulate over long frame sequences. For the freehand mode, we will attempt to detect and eliminate drift using depth enhanced panoramas as a global reference. To avoid the drift in the panorama mode, we will investigate high-dynamic range panoramas, which will eliminate the occasional color inconsistencies visible now and will allow registration of the incoming frame directly with the faces of the cube map.

The next step is to construct complete models of entire rooms. For this we will develop tools for registering and merging partial models acquired in freehand and panorama mode. A longer term goal is to design a version of the system for outdoor use.

7. Acknowledgements

Withheld for double-blind review.

References

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

[Bernardini 02] 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.

[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.

[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, and M. Cohen. The Lumigraph. Proc. of SIGGRAPH 96, 43-54 (1996).

[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).

[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

[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 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.

[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).