The ModelCamera

V. POPESCU, G. BAHMUTOV, E. SACKS, and M. MUDURE
Computer Science, Purdue University

Author names and email addresses:
Voicu Popescu, popescu@cs.purdue.edu (corresponding author)
Gleb Bahmutov, bahmutov@cs.purdue.edu
Elisha Sacks, eps@cs.purdue.edu
Mihai Mudure, mmudure@cs.purdue.edu

Address for manuscript correspondence:
250 N University Street
West Lafayette, IN, 47907-2066
Phone: 765-496-7347
Fax: 765-494-0739

ABSTRACT

The ModelCamera is a system for inside-looking-out modeling of static, room-size indoor scenes. The ModelCamera implements an interactive modeling pipeline based on real-time dense color and sparse depth. The operator scans the scene with a device that acquires a video stream augmented with 49 depth samples per frame. The system registers the frames using the depth and color data, and integrates them into an evolving model that is displayed continually. The operator selects views by checking the display for missing or undersampled surfaces and aiming the camera at them. A model is built from thousands of frames.

Categories and Subject Descriptors: I3.6 [Computer Graphics]: Methodology and Techniques- *Graphics data structures and data types*; I4.1 [Image Processing and Computer Vision]: Digitization and Image Capture- *Camera calibration, sampling, scanning*; I4.8 [Image Processing and Computer Vision]: Scene Analysis- *Range data, stereo, surface fitting*;

General Terms: Design, Algorithms, Measurement, Performance

Additional Key Words and Phrases: scanning; 3D models of rooms; computer vision; computer graphics; structured light; triangulation; user in the loop; interactive feedback; panoramas; texture-mapped triangle meshes; interactive rendering.

1. INTRODUCTION

Three-dimensional models of real world scenes enable important computer graphics applications in science, engineering, education, health care, defense, homeland security, forensics, opinion forming (mass media, courts of law, decision making), cultural heritage preservation, and entertainment.

1. 1. Research problem

Modeling real world scenes manually using CAD or animation software (e.g. AutoCad, 3dsmax, Maya) requires technical training, artistic talent, and a huge time investment. Even so, the resulting models fail to capture the full complexity of real world scenes. The alternative is automated modeling based on the following pipeline. Depth and possibly color data is acquired from several views. Depth is measured actively or is inferred from color. The data is registered in a common coordinate system. The registered data is merged and is encoded in a model suitable for the application. Current automated modeling techniques have disadvantages that rule out many applications.

Long modeling cycle

Depth acquisition takes tens of minutes per view due to sequential high-resolution scanning in laser rangefinding or correspondence searching in depth from stereo. Registration and model construction are slow because the datasets are huge. Model construction has to discard the redundant samples caused by the overlap between views and by the excessive depth sampling of flat surfaces. The lengthy modeling cycle increases the cost of the acquired models.

Lack of support for inside-looking-out modeling

The best results of current automated modeling techniques are obtained in the *outside-looking-in* case where objects are modeled from outside their bounding volume. Examples are scanning a statue for a virtual museum or scanning a piston for reverse

engineering. Outside-looking-in modeling offers the advantages of scene lighting control and of calibrated mechanical positioning of the objects or of the acquisition device.

However, outside-looking-in modeling has two major disadvantages. First, it is expensive since it requires displacing the (possibly heavy or priceless) objects of interest, or building sophisticated mechanical positioning devices. Second, outside-looking-in modeling is simply not an option for many computer graphics applications that need a model of the inside of a scene. For example acquiring a room, a building, or an entire city requires modeling in the *inside-looking-out* case, where the acquisition device is immersed in the scene of interest. The sheer size of the scene to be modeled, the great range of depths and light intensities, and the conundrum of occlusions that has to be elucidated for acceptable coverage make inside-looking-out modeling much more challenging. The reward is proportional: interactively exploring a 3D scene from within cannot be rivaled by any set of high resolution digital stills or video segments.

Lack of robustness

Devices for active depth acquisition are hard to operate and lose calibration easily. Active and passive depth acquisition techniques require special conditions, such as presence or absence of color texture, diffuse surface reflectance, and controlled lighting. The lengthy modeling cycle prevents early detection of calibration and data acquisition problems. By the time the problems are noticed, addressing them is expensive and often impossible.

High cost

In addition to the time cost, some automated modeling techniques (e.g. time-of-flight and triangulation laser rangefinding) require expensive equipment and expert operators. The high cost of the acquired models precludes the widespread use of automated modeling.

1. 2. Our approach

This paper describes the ModelCamera, an automated modeling system for efficient,

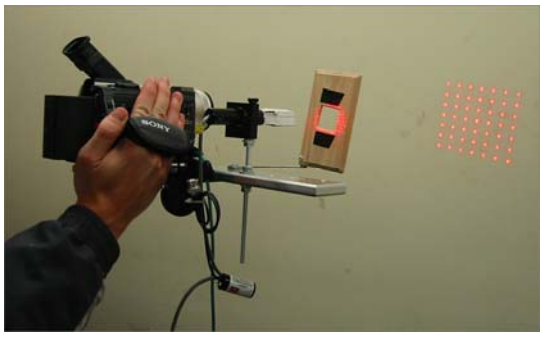


Figure 1 ModelCamera acquisition device: handheld and mounted in parallax-free pan-tilt bracket.

robust, and inexpensive inside-looking-out modeling of static indoor scenes. The ModelCamera implements a novel interactive modeling paradigm that captures structured and unstructured scenes using dense color and sparse depth. We acquire sparse depth quickly, robustly, and inexpensively, and sequences of sparse depth frames offer the same modeling power as dense depth from a few views.

Interactive modeling pipeline

Our interactive modeling pipeline operates as follows. The operator scans the scene with the ModelCamera, which is either handheld or placed in a bracket with two degrees of freedom (Figure 1). The ModelCamera consists of a video camera and a laser system that


Figure 2 *Left*: Real-time freehand modeling of structured scenes; immediate operator feedback shown at top right corner of each image. *Right*: Room fragment modeled in 28s with 133 frames.

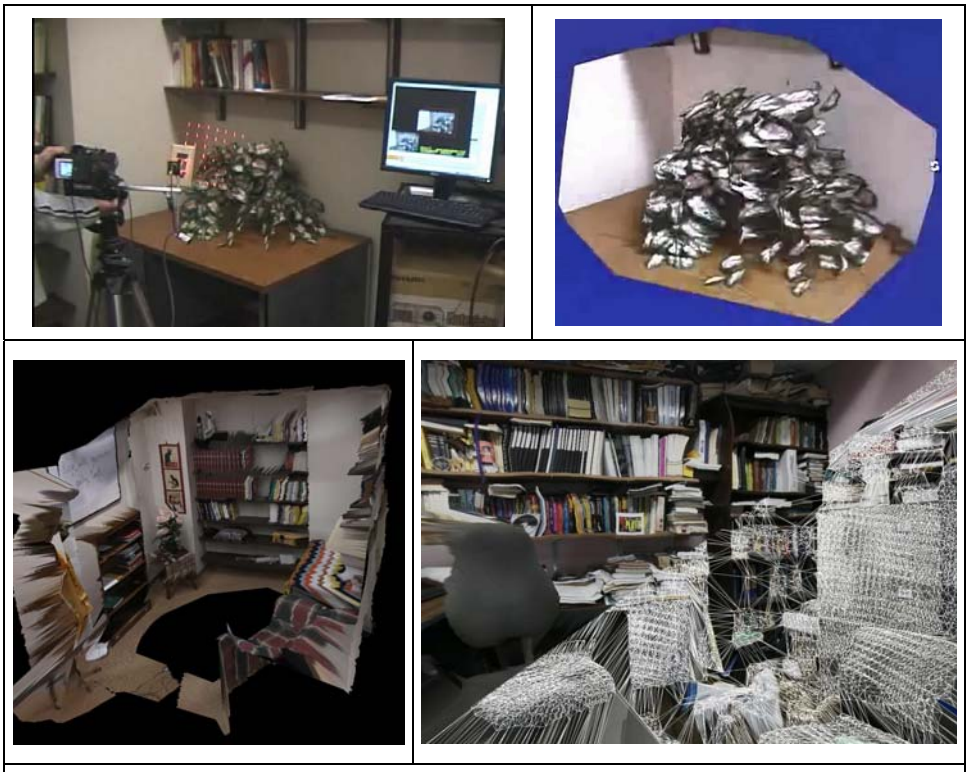

Figure 3 *Top:* Real-time tripod modeling of unstructured scenes with immediate operator feedback, and plant modeled as DEP in 120s from 500 frames. *Bottom*: room DEPs of ~300,000/170,000 triangles acquired in 20/15 minutes from ~6,000/4,400 frames.

casts a 7x7 matrix of beams in the camera's field of view. The video frames are read into a computer. The laser dots are located in each frame and depth samples are computed by triangulation. The video frames (dense color) augmented with 49 depth samples (sparse depth) are registered and integrated into an evolving 3D model. The model is displayed continually. The operator selects views by checking the display for missing or poorly sampled surfaces and aiming the acquisition device at them. A complete model is built from thousands of frames.

Structured vs. unstructured scenes

We distinguish between structured and unstructured scenes. Structured scenes consist of large surfaces such as doors, walls, and furniture (Figure 2). Unstructured scenes contain

many small surfaces, for example a plant, a messy bookshelf, or coats on a rack (Figure 3). The distinction is important for applications and facilitates automated modeling.

While a structured scene can be approximated well with a compact geometric model, achieving the same level of approximation for unstructured scenes requires a model of substantial complexity. Most applications that involve large-scale inside-looking-out scenes do not need and cannot afford highly detailed models for the unstructured sections of the scene. For example, when the goal is to simulate physical phenomena involving an entire building, it is important to generate a practical number of finite elements, so one should not model small sub-volumes at a disproportionate cost. When the goal is rendering, as for example in virtual training of emergency response personnel, a complete model of every leaf of every plant would overwhelm the graphics engine and would interfere with the primary task of providing the trainee with images of the environment at interactive rates. Our approach is to support fast approximate modeling of unstructured scenes, with the option of adding detail at proportional cost.

From the modeling perspective, it makes sense to take advantage of the relative simplicity of the structured sections of the scene, which represent a significant fraction of indoor scenes. Structured scenes do not need the same depth sampling resolution as unstructured scenes. Unlike prior automated modeling techniques, our interactive modeling paradigm gives the operator the information and flexibility to treat the two types of scenes differently and to acquire structured scenes efficiently.

Structured scenes are modeled freehand (the ModelCamera is hand held, see Figure 2). The depth samples of each frame are grouped according to surfaces, depth is interpolated between the samples, and the frame is registered using the interpolated depth and dense color. The registered frames are merged into a model consisting of depth images created on demand.

Our approach to modeling unstructured scenes is to constrain the motion of the ModelCamera. We allow for only two degrees of freedom using a parallax-free camera bracket (Figure 1 *right*, and Figure 3). The sparse depth and dense color frames are registered using color and are merged into a Depth Enhanced Panorama (DEP). DEPs provide a quick and robust method for modeling unstructured scenes. The geometry resolution of a single DEP is improved by merging several DEPs.

Preliminary versions of portions of this work appear in several conference proceedings [Popescu 2003, 2004, Bahmutov 2005]. The remainder of the paper is organized as follows. The next section discusses prior work. Section 3 describes our acquisition device. Section 4 presents our approach for modeling structured scenes. Section 5 describes modeling unstructured scenes. Results are presented for each of the sections. Section 6 provides a discussion of our automated modeling approach and sketches directions of future work.

2. PRIOR WORK

We structure the discussion of prior techniques for automated scene modeling according to how they solve the problem of depth acquisition.

2. 1. Acquired dense depth

Depth from stereo, triangulation laser rangefinding, and time-of-flight laser rangefinding technologies acquire dense, accurate depth maps that can be converted into high-quality models. Examples include digitizing Michelangelo's statues [Levoy 2000, Bernardini 2002], Jefferson's Monticello [Williams 2003], cultural treasures of Ancient Egypt [Farouk 2003], the Parthenon [Stumpfel 2003], and the ancient city of Sagalassos [Pollefeys 2001, 2002].

A 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.

The goal of the work presented in this paper has not been to devise new depth acquisition technology, but rather to achieve interactive inside-looking-out modeling. No dense depth acquisition device is sufficiently fast, compact, robust, and inexpensive, but the problem of depth acquisition continues to be investigated from several directions. Computer vision researchers continually increase the quality and robustness and decrease the acquisition time of depth maps extracted from images, as recently reported in [Darabiha 2003], [Yang 2003], [Zhang 2003], and [Davis 2003]. The Zcam technology [3dvsystems] acquires depth in parallel over the entire view frustum. The Zcam is used as an add-on to studio cameras for real time depth keying [Gvili 2003]. Its depth resolution is too low for modeling. Helmoltz stereopsis [Zickler 2002] is a new depth extraction technique that provides normal and depth estimates—thus combining the advantages of classic and photometric stereo—without restricting the surface reflection model. The problem of quickly finding correspondences remains. Should a device emerge that satisfies the needs of inside-looking-out interactive scene modeling, it will be integrated with our system.

2. 2. User-specified coarse 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 [Hidalgo 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.

2. 3. 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 rapid, inexpensive acquisition and of interactive photorealistic rendering, which makes them the only inside-looking-out modeling technique widely used. Panoramic image mosaics [Szeliski 1996, Zoghiami 1997, Coorg 1998] relax the same-center-of-projection constraint and stitch seamlessly hundreds of images that were taken from the approximately same location.

Omnidirectional imaging enables the direct acquisition of wide-field-of-view 2D data base of concurrent or quasi-concurrent rays using dioptric (e.g. fish eye lens) or catadioptric cameras (e.g. parabolic mirror, planar pinhole camera cluster) [Nayar 1997, Kropp 2000, Aliaga 2001, Geyer 2002]. The disadvantage of the 2D ray database approach is the lack of geometry. This precludes physical simulation applications, and severely restricts computer graphics applications by not supporting view translations and depriving the user of motion parallax, 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. Like panoramas, lightfields do not support applications that require geometry.

2. 4. 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 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.

Some systems obtain frame registration from external trackers (e.g. electromagnetic senders and receivers, or mechanical arms) and concentrate on integrating the depth and color data of each frame into a texture mapped geometric model [Fisher 1996], [Hilton 2000]. 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.

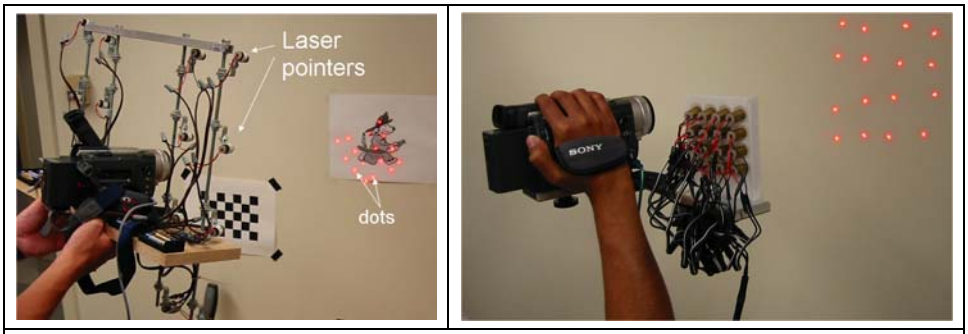

Figure 4 Earlier ModelCamera prototypes.

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.

3. ACQUISITION DEVICE

To enable our interactive inside-looking-out modeling paradigm the acquisition device has to be portable and has to acquire dense color and sparse depth quickly and robustly.

The ModelCamera prototype seen in Figure 1 succeeds two earlier prototypes. The first prototype (Figure 4, *left*), uses 16 laser diodes placed around the video camera. The laser beams quasi-converge at 70cm in front of the camera and then diverge, which allows the

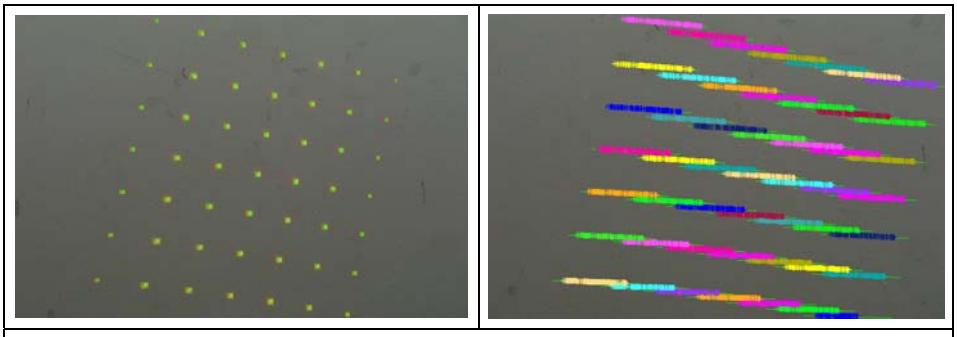

Figure 5 Epipolar line calibration. 49 dots found in one of 200 frames (*left*) and epipolar segments fitted trough each of 49 groups of 200 dots (*right*).

operator to conveniently adjust the spacing between the laser dots by translating the camera back and forth [Popescu 2003]. The second prototype (Figure 4, *right*) sacrifices this capability in favor of improved rigidity by mounting the diodes on a rapid-prototyped plate attached to the camera with an aluminum bracket. The current prototypes replace the diodes with a compact single laser source whose beam is split with a diffraction grating as described below.

3. 1. Hardware

The ModelCamera consists of a digital video camera and a laser system. We use a high-end consumer-level digital video camera that 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. We use an off-the-shelf laser diode and diffraction grating [Stockeryale] that weigh 100g and produce 1mW dots (class IIb, eye safe). 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.

For modeling unstructured scenes the ModelCamera is placed in a bracket adapted from a commercial tripod head for telephoto lenses [Wimberley]. The bracket allows panning and tilting the camera about its center of projection. The two rotation axes of the bracket are coplanar by construction.

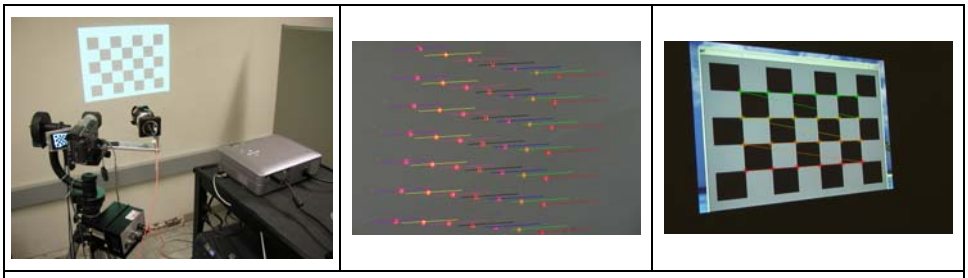

Figure 6 Laser beam 3D calibration. Setup with projected grid (*left*), frame used for dot detection (*middle*) and corresponding frame with projected grid for extrinsic calibration (*right*).

The goal is to place the camera in the bracket such that its center of projection coincides with the intersection of the two axes. We achieve this by trial and error: we slide the camera forward-backward and up-down on the bracket until panning and tilting do not show motion parallax between a near (30cm) and a far (350cm) object. We estimate that the accuracy achieved is better than 5mm since a translation of 5mm away from the calibrated position in either direction introduces clearly noticeable motion parallax when panning or tilting. This figure is confirmed by another indirect measurement: we scan a white wall by panning the ModelCamera. The distance to the wall measured using the depth samples stays within 3mm.

3. 2. Calibration

The ModelCamera is calibrated in three steps. First, the video camera intrinsics including distortion coefficients are found using standard camera calibration [Bouguet]; the calibration error is 0.075 pixels, computed as the average image plane distance between the projection of a 3D point (checkerboard corner) and the location where it was found in the image.

Since the laser beams are fixed with respect to the camera, their image plane projection is confined to a fixed epipolar line. The second calibration step finds the epipolar line of each laser beam. 200 frames are recorded by aiming the camera at a white wall from various distances. The 49 laser dots are found in each undistorted frame. A 2D

line is least-squares fitted through each group of 200 points, with an average error of 0.2 pixels (Figure 5).

The third step finds the 3D line equation for each laser beam in the camera coordinate system. The ModelCamera is placed on a tripod and is aimed at a white wall (Figure 6). The laser dots are found on their respective epipolar lines. Then the laser is turned off and a checkerboard is projected orthographically onto the wall. Using the checkerboard and the known intrinsics the camera pose is found in the coordinate system defined by the checkerboard. A 3D point is found for each laser beam by intersecting the checkerboard plane *xy* with the camera ray where the dot was found. The tripod is moved to other locations to collect several (~10) 3D points for each laser beam. A line is least-squares fitted to the points of each laser beam with an error of 2.5mm, computed as the average distance from the points to the line.

3. 3. Robust sparse depth in real time

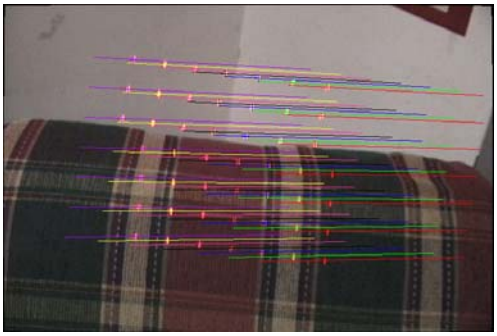

Figure 7 Frame with dots found on the epipolar segments.

Once the ModelCamera is calibrated, each point on each epipolar segment corresponds to a 3D point. The right/left endpoints of the epipolar segments (Figure 7) correspond to a depth of 50/350cm. The diffraction grating and the orientation of the laser head ensure that the epipolar lines are disjoint and as far apart from each other as possible. This avoids dot confusion and makes depth acquisition robust. Each dot is found along its epipolar segment by searching for an intensity peak.

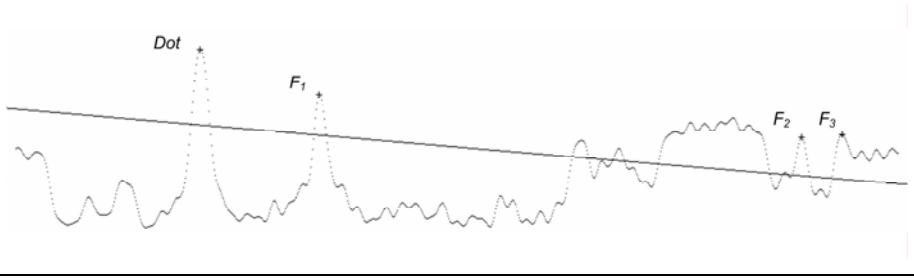

Figure 8 Intensity along epipolar line with dot and false peaks. Line indicates threshold.

The dot detector finds intensity peaks along the epipolar segments (Figure 8). Candidate peaks have to pass additional 2D symmetry tests.

3. 4. Results

Depth accuracy

The average dot detection success rate is between 60% and 99% depending on distance and surface properties. Dot detection takes 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.

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.

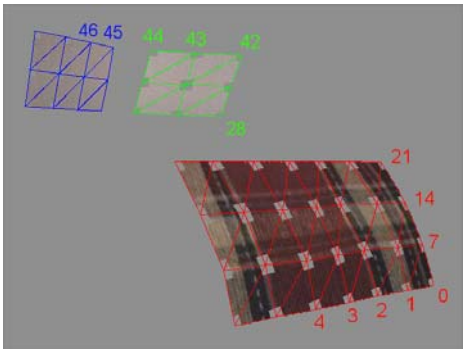

Figure 9 Surface fitting.

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 due to the limited number of views.

4. STRUCTURED SCENES

We acquire structured scenes freehand. Each frame is processed in three steps:

(1) fit the depth samples per frame with polynomial surfaces;

Figure 10 Depth then color registration: pair of frames before depth registration (*left*), after depth registration (*middle*), and after color registration (*right*).

(2) register consecutive frames using surfaces and color data;

(3) merge the registered frames into a scene model.

4. 1. Surface fitting

The dots in a frame are grouped into surfaces. Figure 10 shows the three surfaces fitted through the dots of the frame shown in Figure 7. 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. Surface boundaries appear as depth discontinuities (couch/wall) or as curvature discontinuities (wall/wall). We construct a dot connectivity graph by scanning each row and column of dots for discontinuities. Adjacent dots are connected unless their second divided depth difference exceeds a threshold. We least-squares fit a polynomial surface $z = P(x, y)$ to the dots in each connected component. The dots are assigned surface normals according to the polynomials.

4. 2. Registration

The registration task is to compute the motion of the ModelCamera from frame to frame. Given dense depth, registration is commonly performed with the iterative closest point algorithm [Besl 92, Rusinkiewicz 2002]. This algorithm does not work with sparse depth. Nor does it handle symmetric surfaces (explained below), which are the norm in

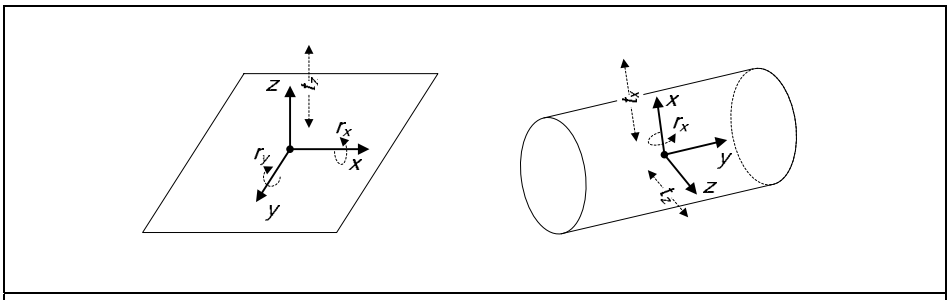

Figure 11 Depth asymmetric motions.

structured scenes. We achieve fast, robust registration by first aligning the depths in the two frames then aligning the colors (Figure 10).

Depth registration

We perform depth registration by least-squares fitting laser dots in the new frame to old frame surfaces. The motion is linearized as $m(p) = t + p + r \times p$ with $t = (t_x, t_y, t_z)$ the translation and $r = (r_x, r_y, r_z)$ the angular velocity. An equation is generated for each dot that lies on the largest surface in both frames. The old surface is linearized as $n(c-a) = 0$

with n the surface normal, c the new dot, and a the old dot. The depth equation is $tn + r(c \times n) = n(a-c)$.

The 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 system has a unique least-squares solution when A has rank six. In structured scenes, the 49 dots provide ample equations, but symmetric surfaces generate linearly dependent equations. A surface is symmetric when it is invariant under translation along an axis or rotation about an axis. Examples are planes, surfaces of extrusion, surfaces of rotation, and spheres. The distance from the dots to the surface is constant when the camera performs these motions, so they cannot be computed from any amount of depth data.

We restrict the depth equations to a 3-dimensional subspace of x that represents asymmetric motion. To identify this subspace, we classify the surface as cylindrical, spherical, or planar. We choose a coordinate system whose origin is the centroid of the surface, whose z axis is the surface normal, and whose x and y axes are the major and minor principal axes (Figure 11). If the two principal curvatures are positive and are equal within a tolerance, the surface is classified as spherical and the motions assigned to depth registration are t_x, t_y, t_z. If one curvature is positive and the other is zero, the surface is classified as cylindrical and the motions are t_x, t_z, and r_x. Otherwise the surface is classified as planar with motions t_z, r_x, and r_y. This case covers planes and asymmetric surfaces, which are well approximated by their tangent planes.

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

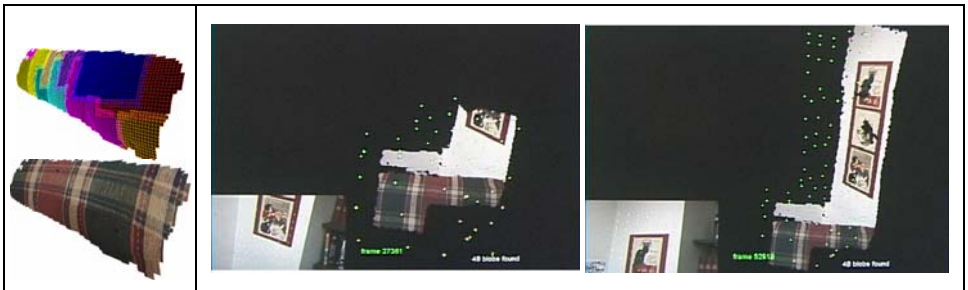

Figure 12 *Left*: Couch surface modeled with set of depth images (wireframe and texture). *Right*: 2 snapshots of the operator feedback window (see also Figure 2). Green dots show the placement of depth images.

reconstruction is unnecessary for error evaluation, so this approach does not incur the full cost of IBR by 3D warping [Popescu 2003].

4. 3. Model construction

The scene is modeled as a collection of images with per pixel depth (depth images). The depth images are created on demand as scanning progresses (Figure 12). We use depth images because they can be transformed and merged efficiently [Shade 1998, Popescu 2000, 2003]. Each registered frame is processed as follows. The region spanned by the dots is triangulated and transformed into a depth image by assigning a depth value to each color pixel in the region using the triangulation. The depth image of the new frame is merged into the depth images of the model. The better samples are retained. The quality metric is based on the sampling rate. Samples that project at the border between two depth images are repeated to provide overlap. The depth images are rendered efficiently as texture-mapped meshes to provide modeling 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.

4. 4. Results

We have tested the structured scene modeling pipeline 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.

The average/maximum model construction times were 60/120ms. We used 256 x 256 depth images and a triangulation step of 8 pixels, which yields 2K triangles and 256 KB of texture per depth image. Current graphics hardware can easily handle 100 depth images, which is sufficient to show the model obtained from several sequence of frames.

5. UNSTRUCTURED SCENES

5. 1. Registration

Color registration

Unstructured scenes are modeled with the parallax-free camera bracket. Since there is no camera translation, the frames acquired using the bracket can be registered from the color data only, the same way same-center-of-projection images are stitched together to form panoramas [Chen 95]. The rotations for each frame are computed by minimizing the color difference at the region of overlap between frames. For efficiency, only a subset of the color data of each frame is used. The color registration algorithm is similar to the one described in Section 4. 2. A notable difference is that the mapping from one frame to another is now projective texture mapping, which replaces the costlier 3D warping.

Axes calibration

Any two frames with the same center of projection can be registered by finding three

Figure 13 (*Left*) DEP splatting. (*Right*) Magnified fragment with highlighted splat boundaries.

rotations about arbitrary orthogonal axes. However, knowing the two axes of the bracket improves the color registration performance by reducing the search to two degrees of freedom. To allow the operator to begin scanning from any tilt position, the pan axis of the bracket needs to be calibrated for every scanning sequence. We do this by requiring the operator to only pan for 15 degrees at the beginning of a sequence. The frames are registered using color in 3 degrees of freedom and the pan axis is computed as the rotation axis of this motion. The tilt axis is computed similarly, but only once, since it does not change from sequence to sequence. Once the axes are known, the operator can pan and tilt freely, and frames are registered by searching for the two rotations.

5. 2. Model construction: depth enhanced panorama

The registered concentric dense color and sparse depth frames are merged into a Depth Enhanced Panorama (DEP) on the fly. The color data of each frame minus the regions covered by the laser dots is merged into a cube map. The faces of the cube map are subdivided in tiles for efficiency. The current frame updates only tiles that fall within its field of view and are not yet filled. The geometry of the DEP is represented in one of two ways.

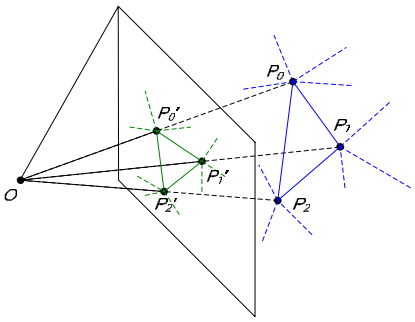

Figure 14 DEP triangulation. One face of a DEP with center O is shown. The depth samples P_0, P_1, and P_2 are connected in a triangle obtained by Delaunay triangulating their projections onto the face.

Disconnected representation

In the disconnected representation the geometry of the DEP is stored as a set of splats. The depth samples accumulated from the individual frames are stored in quadtrees for fast access to the nearest neighbor for a given sample. A quadtree is defined for each face of the cube map. The approach is similar to techniques developed in point-based modeling and rendering: QSplats [Rusinkiewicz 2000], surfels [Pfister 2000], and forward rasterization [Popescu 2000]. None of these methods can be used directly since DEPs are only sparsely populated with depth samples.

A splat is defined for each depth sample. The splat size and normal are derived from the neighboring depth samples. Only neighbors with depth similar to the depth of the current sample are considered. The neighbors are triangulated and the normals of the triangles are averaged to obtain the splat normal. The splat size is the average distance from the depth sample to its neighbors. This fills most of the gaps that would otherwise appear due to the sparse set of depth samples in the DEP. The splats are texture mapped using the cube map faces (Figure 13). The main advantage of the disconnected DEP representation is convenient merging of DEPs, see Section 5. 3. Better visual quality is obtained when the depth samples are connected in a triangle mesh.

Figure 15: DEP and magnified fragment with wireframe.

Connected representation

A connected representation of the DEP is obtained by triangulating the depth samples in 2D on the faces of the cube map. A 3D triangle mesh is obtained by applying the connectivity information so obtained to the depth samples (Figure 14). The 2D triangulation is possible since the DEP has a single center of projection, which prevents occlusion and enables an unambiguous 2D ordering of the depth samples. The 3D mesh is texture mapped with the cube map faces (Figure 15). The 2D mesh is triangulated incrementally during the acquisition to accommodate the depth samples of the newly integrated frame. We use a Delaunay tree with logarithmic expected insertion time [Devillers 1992].

5. 3. Multiple depth enhanced panoramas

If the desired view is close to the acquisition viewpoint, a single DEP produces high-quality images of the scene. If the desired view is considerably different from the DEP acquisition view, the image quality degrades because of missing and undersampled surfaces (Figure 16). A wider range of views is supported by using several DEPs of the same unstructured scene.

Figure 16 Plant DEP rendered from views that are near (*left*) and far (*right*) from the acquisition view.

The operator builds the first DEP as before, examines it for missing or poorly sampled surfaces, moves the ModelCamera to a second viewpoint, and starts building the second DEP. Once sufficient depth samples are acquired, the second DEP is registered with the first using three operator-specified point correspondences between the two DEPs. The system computes the rigid camera motion between the two DEPs, the acquisition of the second DEP resumes, and the two DEPs are visualized together.

Disconnected representation

The disconnected representation directly supports multiple DEP rendering without modification. The already completed DEPs and the evolving DEP are continually rendered independently from each other in the splatting mode to guide the operator in completing the model (Figure 17).

Figure 17: *Left*: single DEP shown with splats highlighted in blue. *Middle*: second DEP fills in missing surfaces. *Right*: splats of second DEP highlighted in red.

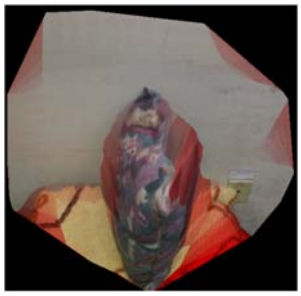

Figure 18 Sampling rate visualization.

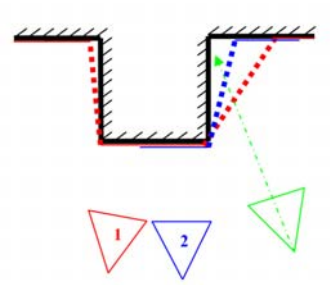

Figure 19 Sample selection.

Connected representation

Straightforward z-buffering multiple DEPs in the connected representation suffers from severe artifacts since better sampled surfaces are often obscured by the worse sampled ones (Figure 20, *middle row*). We render the triangle meshes of multiple DEPs using an algorithm that selects the best parts of the individual meshes (Figure 20).

First, during a preprocessing step, we compute the sampling rate of each triangle as the inverse of the average length of its sides, normalized to 0-1 range (see Figure 18, where undersampled regions are highlighted in red). Triangles with sampling rates below a given threshold are labeled as undersampled. The threshold is established experimentally for each scene.

Then, during runtime, we select the three DEPs with acquisition viewpoints closest to the desired viewpoint. Each input DEP is rendered from the desired view into a separate high-resolution 2048x2048 pixel OpenGL color and z buffer. The sampling rate of the triangle is stored in the alpha channel. The separate color and z buffers are bound as texture maps and combined on per pixel basis using a fragment program. If a pixel is covered in at least one DEP by a triangle that is not undersampled, the closest such triangle is used. If an output pixel is covered by undersampled triangles in all of the DEPs, the algorithm selects the farthest undersampled surface. In Figure 19 the desired

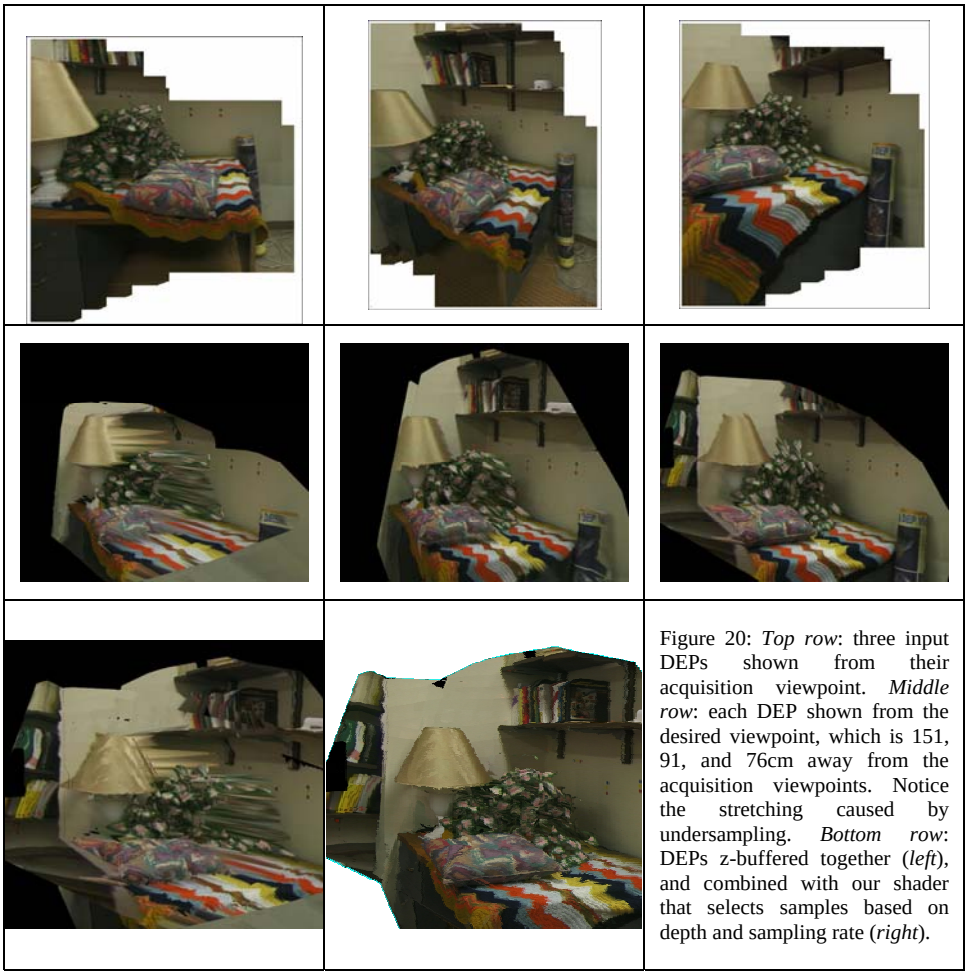

Figure 20: *Top row*: three input DEPs shown from their acquisition viewpoint. *Middle row*: each DEP shown from the desired viewpoint, which is 151, 91, and 76cm away from the acquisition viewpoints. Notice the stretching caused by undersampling. *Bottom row*: DEPs z-buffered together (*left*), and combined with our shader that selects samples based on depth and sampling rate (*right*).

view (green) ray intersects undersampled triangles in DEPs 1 and 2. We select the farthest sample coming from DEP 2. The farther sample invalidates the triangle coming from DEP 1, and indicates that DEP 2 approximates the scene more closely. This heuristic works well in practice (Figure 20) and produces better results than selecting the undersampled surface with the highest sampling rate.

5. 4. Results

DEPs offer an efficient and robust method for modeling unstructured scenes, which are challenging for any automated modeling technique. The color registration algorithm is fast (average running time 150ms per frame) and, on average, only fails once in 100

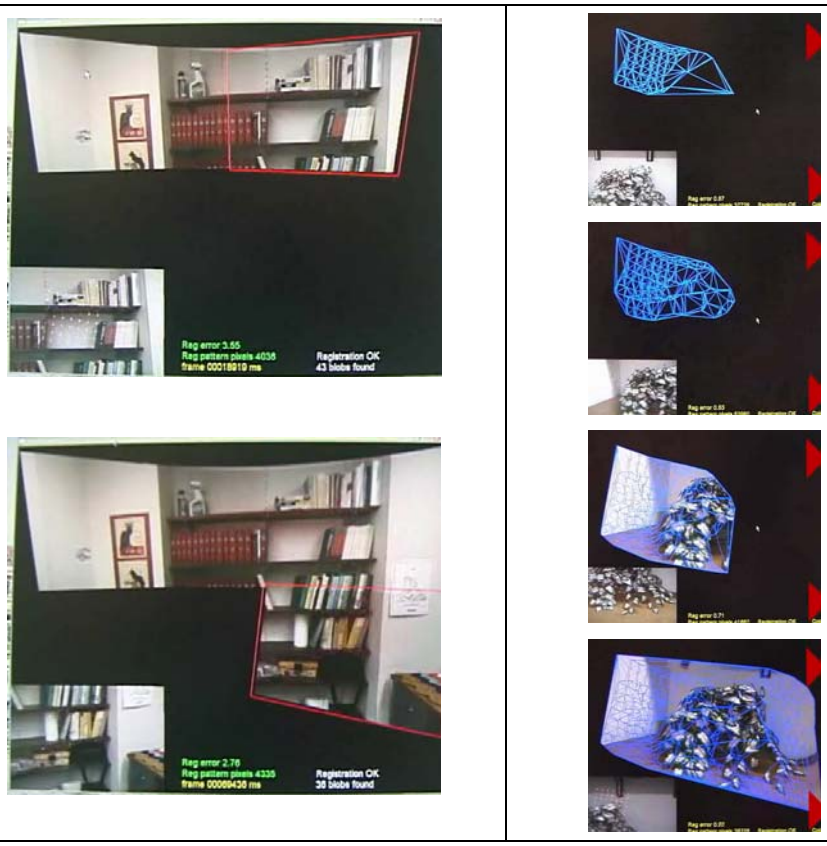

Figure 21 Operator feedback. (*Left*) Red frame marks last registered frame, current frame is shown at the bottom left corner of the feedback window. *(Right)* DEP is built incrementally and visualized in real time.

frames. The immediate operator feedback allows the operator to easily regain registration by aligning the camera view with the last registered frame (Figure 21, *left*). Incrementing the DEP with the contribution of the current frames takes on average 50ms, for an overall average frame rate of 5 fps.

The camera field of view is approximately 40 degrees, and, at 5fps, the color map is filled in quickly. The model is available immediately (Figure 21, *right*) which allows the operator to add geometric detail where desired by revisiting the regions with high complexity. The additional frames contribute new depth samples which refine the DEP.

The possibility of adding detail at linear cost is an important advantage of our interactive modeling pipeline.

To support viewpoints that are far away from the acquisition point we acquire several DEPs of the same scene. The performance of the GPU program that merges at run time the three best DEPs depends on two main factors: the number of primitives in each input DEP and the target frame buffer resolution. The per pixel merging of the individual OpenGL color and depth buffers is sped up by using the WGL_ARB_pbuffer and GL_ARB_multitexture extensions [SGI], which allow drawing into and combining multiple rendering contexts on the graphics card without transfers of the pixel buffers to/from main memory. We have achieved 5 fps rendering rate using the nVidia Quattro FX 3000 graphics card for a 512x512 output frame buffer and three input DEPs with more than 40K triangles each. The computation of the triangle sampling rate as preprocess takes less than a second.

6. DISCUSSION

We have presented a novel approach to inside-looking-out modeling of static scenes based on dense color and sparse depth. Fast data acquisition accelerates the modeling pipeline to interactive rates, which allows the operator to guide the modeling process in real time. We have designed and built a structured light indoor acquisition device that robustly acquires video frames augmented with 49 depth samples, and we have developed algorithms for registering and integrating sequences of such frames into models of structured and unstructured scenes. The ModelCamera avoids by design the difficult problem of identifying the light pattern elements. The device acquires in a single pass high-quality color intrinsically registered with depth. Other advantages of our design are portability, intuitive operation and small cost.

Compared to triangulation and time-of-flight laser rangefinders, the ModelCamera produces models of lower fidelity, but only requires a fraction of the time and equipment cost. Compared to color panoramas, our method shares their advantages of efficient modeling, intuitive operation, and low equipment cost, and removes their fundamenta, disadvantage by allowing viewpoint translation.

We will continue to extend our system. We will develop tools for assembling complete room models from structured and unstructured scene segments, and for aligning several rooms and corridors in complete building models relying on constraints generated by shared walls, floors and ceilings. Corridors are important for many applications that require 3D models of building interiors so we will address them as a special case. Because of their relatively simple geometry corridors do not need to be scanned on their entire length. We will acquire DEPs at each corner and extend the side wall, ceiling, and floor planes.

In this work we have handled the difficult case of unstructured scenes by limiting the degrees of freedom of the ModelCamera motion to two parallax-free rotations. We will research removing this limitation. The first step is to add a translational degree of freedom, which will be provided by an extension to the camera bracket. The ultimate goal is to acquire unstructured scenes freehand. We will investigate accelerating and improving existing pose estimation algorithms that rely on feature tracking by taking advantage of the depth samples our system acquires at each frame.

7. ACKNOWLEDGMENTS

We are grateful to the members of our graphics group for numerous useful discussions. This research was supported by NSF grants CCR-9617600 and SCI-0417458, by the Purdue Computer Sciences Department, and by Purdue's Visualization Center.

8. REFERENCES

3DV Systems www: http://www.3dvsystems.com

ALIAGA, D., AND CARLBOM, I. 2001, Plenoptic Stitching: A Scalable Method for Reconstructing 3D Interactive Walkthroughs. In *Proceedings of SIGGRAPH '01,* Los Angeles, CA, 2001.

ANJYO, K., HORRY, Y., AND ARAI, K. 1997. Tour into the Picture. In *Proceedings of SIGGRAPH '97*, Los Angeles, CA, 1997.

BAHMUTOV, G., et al 2005. Depth Enhanced Panoramas. In proc. of Second International Conference on Video, Vision, and Graphics. Edinburgh, July 2005.

BERNARDINI, F., et al. 2002. Building a Digital Model of Michelangelo's Florentine Pieta. *IEEE Computer Graphics & Applications*, Jan/Feb. 2002, 22(1), pp. 59-67.

BESL, P., AND MCKAY, N. 1992. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2) (1992) 239-256.

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

BOUGUET, J.-Y. AND PERONA, P. 1999. 3D Photography using Shadows in Dual-Space Geometry. *International Journal of Computer Vision,* Vol. 35, No. 2, 1999, pp. 129-149.

BOUGUET, J.-Y. 2005. http://www.vision.caltech.edu/bouguetj/calib_doc

CHEN, S. 1995. Quicktime VR - An Image-Based Approach to Virtual Environment Navigation. In *Proceedings of SIGGRAPH '95*, Los Angeles, CA, 1995.

COORG, S., MASTER N., AND TELLER, S. 1998. Acquisition of a Large Pose-Mosaic Dataset. In *Proceedings of Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 1998.

DARABIHA, 2003. Video-Rate Stereo Depth Measurement on Programmable Hardware. In *Proceedings of IEEE Conference of Computer Vision and Pattern Recognition*, Madison. WI, June 2003.

DAVIS, J., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2003. Spacetime stereo: a unifying framework for depth from triangulation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, Madison. WI, June 2003.

DEBEVEC, P., TAYLOR, C., AND MALIK, J. 1996. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image Based Approach. In *Proceedings of SIGGRAPH '96,* New Orleans, LA, 1996.

DEVILLERS, O. MEISER, S. AND TEILLAUD, M. Fully dynamic Delaunay triangulation in logarithmic expected time per operation. *Comput. Geom. Theory Appl.*, 2(2):55-80, 1992.

FAROUK, M. et al. 2003, "Scanning and Processing 3D Objects for Web Display". *In Proceedings of 4th International Conference on 3D Digital Imaging and Modeling (3DIM '03)*, Banff, Canada, October 2003.

FISHER, R., FITZGIBBON, A., GIONIS, A., WRIGHT, M., AND EGGERT, D. "A Hand-Held Optical Surface Scanner for Environment Modeling and Virtual Reality" In Proc. Virtual Reality World, Stuttgart, Germany, pages 13-15 1996.

GEYER, C. AND DANIILIDIS, K. 2002. Omnidirectional Video, *The Visual Computer*, (2002).

GORTLER, S. et al. 1996. The Lumigraph. In *Proceedings of SIGGRAPH '96*, New Orleans, LA, 1996.

GVILI, R., KAPLAN, A., OFEK, E., AND YAHAV, G.. "Depth Key" SPIE Electronic Imaging 2003 Conference Santa Clara, California.

HEBERT, P. 2001. A self-referenced hand-held range sensor, In *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, Quebec, Canada, 2001.

HIDALGO, E., AND HUBBOLD, R. J. 2002. Hybrid geometric-image-based-rendering. In *Proceedings of Eurographics 2002*, Computer Graphics Forum, 21(3):471-482, September 2002.

HILTON, A. AND ILLINGWORTH. “Geometric Fusion for a Hand-Held 3D Sensor” 12(1)::44-51m, Machine Vision Applications, 2000.

KONINCKX, T. P., GRIESSER, A., AND VAN GOOL, L. 2003. Real-Time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light. In *Proceedings of Fourth International Conference on 3D Digital Imaging and Modeling*, Banff, Canada, 2003.

KROPP, A., MASTER, N., AND TELLER S. 2000. Acquiring and Rendering High-Resolution Spherical Mosaics. In *Proceedings of IEEE Workshop on OmniDirectional Vision*, Hilton Head Island, SC, June 2000.

LEVOY, M. et al. 2000. The Digital Michelangelo Project: 3D Scanning of Large Statues, In *Proceedings of SIGGRAPH ’00*, New Orleans, LA, 2000.

LEVOY, M., AND HANRAHAN, P. 1996. Light Field Rendering. In *Proceedings of SIGGRAPH ‘96*, New Orleans, LA, 1996.

MCMILLAN, L. AND BISHOP, G. 1995. Plenoptic modeling: An image-based rendering system. In *Proceedings of SIGGRAPH '95*, Los Angeles, CA, 1995.

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

NAYAR, S. 1997. Catadioptric Omnidirectional Cameras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.

OH, B. M. et al. 2001. Image-Based Modeling and Photo-Editing. In *Proceedings of SIGGRAPH ’01*, Los Angeles, CA, 2001.

PFISTER, H. et al. 2000. Surfels: Surface Elements as Rendering Primitives. In *Proceedings of SIGGRAPH '98*, Orlando, FL, 1998.

POLLEFEYS, M. AND VAN GOOL, L. 2002. From Images to 3D Models*, Communications of the ACM*, July 2002/Vol. 45, No. 7, pp.50-55.

POLLEFEYS, M. et al. 2001. "A Guided Tour to Virtual Sagalassos", In *Proceedings of VAST2001 (Virtual Reality, Archaeology, and Cultural Heritage),* Athens, Greece, November 2001*.*

POPESCU, V., SACKS, E., AND BAHMUTOV, G. 2004. Interactive Modeling from Dense Color and Sparse Depth*.* In *Proceedings of Second International Symposium on 3D Data Processing, Visualization, and Transmission*. Thessaloniki, Greece, September 2004.

POPESCU, V., SACKS, E., AND BAHMUTOV, G. 2003. The ModelCamera: A Hand-Held Device for Interactive Modeling. In *Proceedings of Fourth International Conference on Digital Imaging and Modeling*, Banff, Canada, 2003.

POPESCU, V. et al. 2000. The WarpEngine: An Architecture for the Post-Polygonal Age. In Proceedings of SIGGRAPH ’00, New Orleans, LA, 2000.

RUSINKIEWICZ, S. AND LEVOY, M. 2000. QSplat: A Multiresolution Point Rendering System for Large Meshes. In *Proceedings of SIGGRAPH ’00*, New Orleans, LA, 2000.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-Time 3D Model Acquisition. In *Proceedings of SIGGRAPH ’02*, San Antonio, TX, 2002.

SEITZ S. M. AND DYER C. R. View Morphing Proc. SIGGRAPH 96, 1996, 21-30.

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

SHADE, J. et al. 1998. Layered Depth Images, In *Proceedings of SIGGRAPH ‘98*, Orlando, FL, 1998.

STOCKERYALE, http://www.stockeryale.com/

STUMPFEL, J. et al. 2003. Digital Reunification of the Parthenon and its Sculptures. In *Proceedings of 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, Brighton, UK, 2003.

SZELISKI, R. Video Mosaics for Virtual Environments. In IEEE Computer Graphics and Applications, (1996).

TAKATSUKA, M. et al. 1999. Low-cost Interactive Active Monocular Range Finder, in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, (1999).

WILLIAMS, N. et al. 2003. Monticello Through the Window. In *Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST 2003)*, Brighton, UK, November, 2003.

WIMBERLEY, http://www.tripodhead.com/index.cfm

YANG, R. AND POLLEFEYS, M. 2003, Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware, In *Proceedings of Conference on Computer Vision and Pattern Recognition*, Madison. WI, June 2003.

ZHANG, L., CURLESS, B., AND SEITZ, S.M. 2003. Spacetime stereo: shape recovery for dynamic scenes. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, Madison. WI, June 2003.

ZICKLER, T. E., BELHUMEUR, P. N. KRIEGMAN, D. J. Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction. International Journal of Computer Vision 49(2/3), 215–227, 2002, 2002 Kluwer Academic Publishers.

ZOGHIAMI I., FAUGERAS O., AND DERICHE R. 1997. Using Geometric Corners to Build a 2D Mosaic from a Set of Images. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.