

Immersive Electronic Books for Surgical Training

Greg Welch, Andrei State, Adrian Ilie, Kok-Lim Low, Anselmo Lastra, Bruce Cairns, Herman Towles, and Henry Fuchs
University of North Carolina at Chapel Hill

Ruigang Yang
University of Kentucky

Sascha Becker, Daniel Russo, Jesse Funaro, and Andries van Dam
Brown University

Immersive electronic books (IEBooks) for surgical training will let surgeons explore previous surgical procedures in 3D. The authors describe the techniques and tools for creating a preliminary IEBook embodying some of the basic concepts.

Surgical training has traditionally followed the adage, “See one, do one, teach one,” but the complexity and rarity of some procedures limits a surgeon-in-training’s ability to learn them. In trauma-related procedures, for example, surgeons must quickly stabilize the patient, and teaching is secondary to patient care. Such procedures are unplanned, and some critical traumas are rare.

This issue is not trivial. For most surgical specialties, physicians require at least five years of postmedical school training before they are eligible for board certification, and it is frequently many years after board certification before most surgeons are entirely comfortable with their craft. In addition, as in many other technology-driven fields, practicing surgeons must keep up-to-date with a dizzying array of new devices and procedures designed to improve care. Combined with an overall increased focus on limiting medical errors is an intense interest in improving and maintaining surgical competency. Currently, there is no educational environment that can

replace the apprenticeship environment of the operating room. Schools and hospitals can use some teaching modules to teach and assess various technical skills, but there is nothing similar to the flight simulator for pilot training that can replace being in the operating room (see the “Visualizing Procedures in 3D” sidebar).

In 1994, Fuchs et al. outlined a vision and some methods for using a “sea of cameras” to achieve 3D telepresence for remote medical procedures.¹ We have since pursued related work formally and informally with multiple collaborators. In 2002, we presented an intermediate summary of results from a three-year effort incorporating teleimmersion technologies into surgical training.² Our goal is to create technologies that let surgeons witness and explore a previous surgical procedure in 3D as if they were present at the place and time of the actual event, with the following added benefits: nonlinear control over time; instruction from the original surgeon or other instructor; and integrated 3D illustrations, annotations, and relevant medical metadata. This would let surgeons integrate the technical aspects of the procedure with the anatomy, radiography, and physiology of the patient’s condition, thus best simulating and in many ways augmenting the real-time educational experience of learning how to perform an operative procedure. Figure 1 shows an early artist’s sketch of our immersive electronic book (IEBook) and a screen shot from our current prototype.

Here we present a current look at our efforts, describing a set of tools that together comprise a system for creating and viewing an IEBook and displaying some results.

Equipment

An understanding of the physical apparatus involved should help readers better understand how the system lets users create and view content.

Acquisition

The primary piece of acquisition equipment is the camera cube shown in Figure 2. The unit is constructed from rigid modular aluminum framing manufactured by 80/20 (<http://www.8020.net>). Each side is approximately one meter.

On the unit are four fluorescent high-frequency linear lights from Edmund Industrial Optics, and eight 640 × 480 Dragonfly FireWire (IEEE 1394) color cameras from Point Grey Research (PGR). The lights reduce specular reflections and provide even illumination. The cameras support

Visualizing Procedures in 3D

The most obvious substitute for being in the operating room—conventional 2D video—has long been available, but surgeons universally consider these videos marginally effective at best for several reasons:

- Video relegates the viewer to the role of passive observer who cannot interactively change the view.
- Subtle, complex motions that can be critical in an operation are easy to miss.
- Video provides few depth cues and offers only linear control over the playback's timing.
- Replaying video only provides the same experience a second time, rather than permitting a new or changed perspective.

In short, watching a video is not much like experiencing the procedure. Video is data from the procedure, but as Davis points out, "experiences are not data."¹ Davis describes experience as "an intangible process of interaction among humans and the world that has its existence in human minds," noting that we can only capture and archive the materials (data) that "occasion experiences in human minds."¹

Three-dimensional computer graphics systems let sighted people see real or virtual models with the depth, occlusion relationships, and motion parallax they are used to in everyday life. When coupled with interaction devices and techniques, such systems can indeed occasion personal experiences in human minds. Several disciplines have leveraged the power of this par-

adigm in training for many years. Arguably the most successful example is flight simulation for pilot training.² Today flight simulators are considered so effective (and cost-effective) that pilots sometimes use them to train on a new version of a plane, and then fly the real plane for the first time on a regularly scheduled flight with regular passengers.

Medical training and education programs have also used 3D graphics techniques.³⁻⁵ Previous training efforts have primarily combined realistic models with interaction devices (sometimes haptic) to simulate the experience of performing a particular procedure. Rather than aiming to simulate the "do one" of the adage, "see one, do one, teach one," this work aims to improve the opportunities to "see one," and improve the associated learning experience's effectiveness.

References

1. M. Davis, "Theoretical Foundations for Experiential Systems Design," *Proc. ACM SIG MultiMedia 2003 Workshop on Experiential Telepresence (ETP 2003)*, ACM Press, 2003, pp. 45-52.
2. B.J. Schachter, "Computer Image Generation for Flight Simulation," *IEEE Computer Graphics and Applications*, vol. 1, no. 4, Oct. 1981, pp. 29-68.
3. J.F. Brinkley, "The Digital Anatomist Project," *Proc. Am. Assoc. Advancement of Science Ann. Meeting*, AAAS Press, 1994, p. 6.
4. R. Zajchuk and R.M. Satava, "Medical Applications of Virtual Reality," *Comm. ACM*, vol. 40, no. 9, 1997, pp. 63-64.
5. A. Liu et al., "A Survey of Surgical Simulation: Applications, Technology, and Education," *Presence: Teleoperators and Virtual Environments*, vol. 12, no. 6, 2003, pp. 599-614.

FireWire-based synchronization when used with a PGR sync unit, which ensures that all eight cameras image simultaneously, eliminating the need for temporal rectification prior to reconstruction of dynamic 3D models from the 2D images, as we describe in the next section.

To accommodate the FireWire bandwidth for 15-frame-per-second acquisition, we configured the cameras into four camera-pair groups. Each

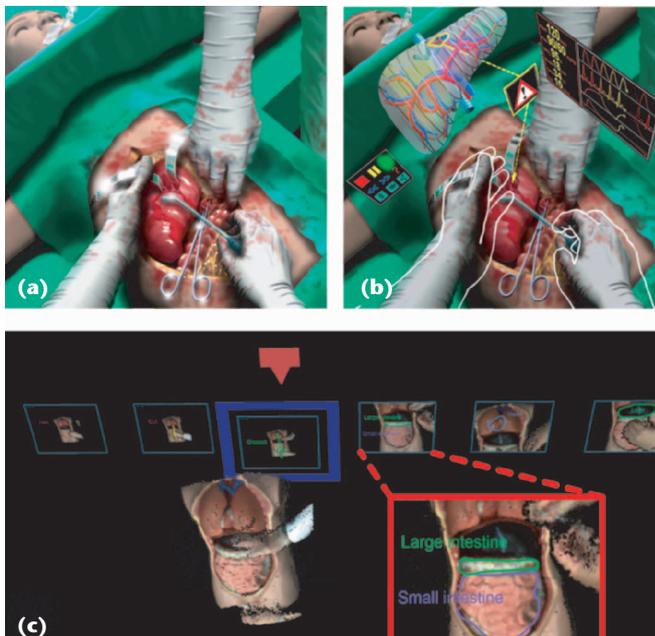
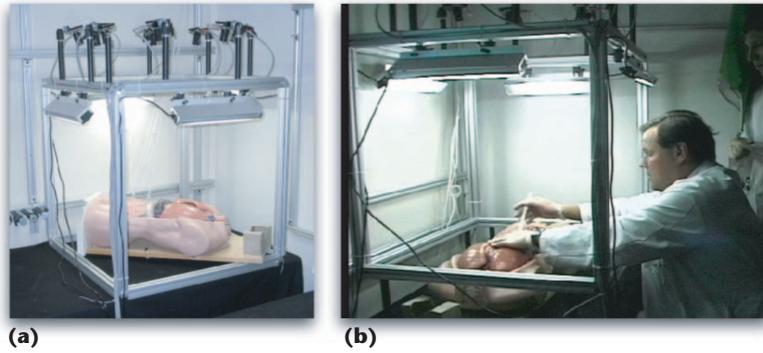


Figure 1. Immersive electronic book (IEBook) in surgical training: (a) early conceptual sketch of a blunt liver trauma surgery, (b) the acquired 3D scene of the surgery with annotations and trainee's hands (sketches by Andrei State); and (c) an IEBook in our present immersive display environment. The third of six annotated snapshots is selected and playing. The red box magnifies the snapshot for the fourth item.

Figure 2. Content acquisition: (a) camera cube used for acquisition with a medical training model in place, and (b) a surgeon (coauthor Bruce Cairns) performing a mock procedure to manage severe blunt liver trauma.



group includes two PGR cameras, a FireWire hub, a PGR sync unit, and a Dell PowerEdge rack-mounted server running Microsoft Windows 2000 Professional. Figure 3 depicts the FireWire connections within a group and between groups.

Reconstruction

For reconstructing 3D models from the 2D images, we use a standard personal computer running Microsoft Windows XP Professional. We currently use a Dell Precision 340 with a 2.2-GHz Pentium 4 and 2 Gbytes of RAM.

Authoring

Both content authoring and viewing take place in a Barco TAN VR-Cube located at Brown University. The VR-Cube, which is approximately three meters on each side, has four projection surfaces (three walls and a floor) and four corresponding Marquee 9500LC projectors configured for a 1,024 x 768 screen with 60-Hz field-sequential stereo. The system uses five commodity Linux workstations: one to gather device data and four to render. The rendering workstations are dual 2-GHz Intel Xeons with genlocked nVidia Quadro FX 3000G graphics cards. Myricom Myrinet connects the workstations for high-speed communication. We use CrystalEyes active liquid crystal shutter glasses, and

Polhemus FasTrak and Intersense IS900 tracking systems.

The primary authoring input device is a handheld Tablet PC. We currently use a Toshiba Portege M200 to navigate the dynamic 3D points/meshes in time, initiate snapshots, highlight and

annotate, and arrange snapshots hierarchically.

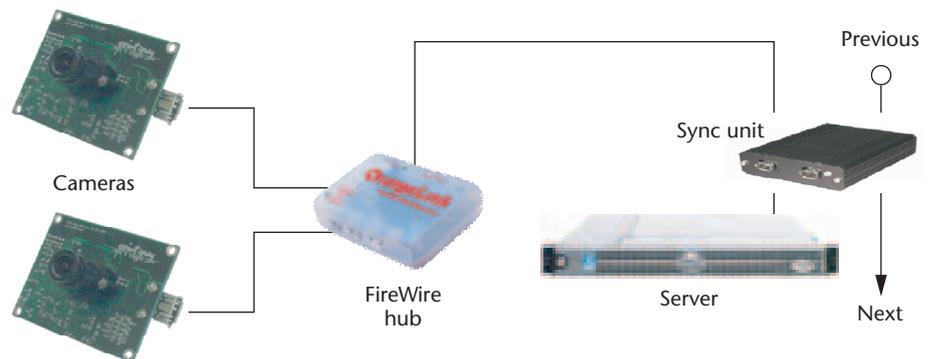
Viewing

Our primary system for fully immersive interaction with a complete IEBook is a combination of the Barco TAN VR-Cube projection system and the Acer Tablet PC. This equipment lets users enjoy both full immersion and the complete set of IEBook features available. This is currently the only viewing option providing a complete experience.

We also explored a hybrid display system combining head-mounted and projector-based displays to simultaneously provide a high-quality stereoscopic view of the procedure and a lower-quality monoscopic view for peripheral awareness (see the “Visualizing Procedures Up Close and All Around” sidebar).

We also support a relatively limited but more accessible Web-based paradigm. The paradigm is limited in that it provides reduced or no immersion, and we currently only show dynamic 3D points/meshes. We use VRML 97 to support Web-based viewing of the dynamic 3D points/meshes with viewpoint control. The VRML data sets use embedded JavaScript functions to implement the user interface actions. The functions are compatible with VRML plugins such as the Cortona plug-in (available at

Figure 3. A single camera-pair group, and the FireWire intra- and intergroup connections. The four groups are connected from previous to next.



Visualizing Procedures Up Close and All Around

Clearly, a student trying to understand a new surgical procedure would appreciate a high-quality, up-close view of the procedure. But what is not so obvious is that learning how to interact and deal with events and confusion around the patient and in the room is also critical. A display with high-fidelity stereo imagery at arm's length and a wide (surrounding) view field can engender this complete experience. Head-mounted displays (HMDs) can provide the former, and projector-based displays the latter, but unfortunately we are not aware of any display system that simultaneously meets both needs.

We merged the two paradigms and developed a hybrid display system, shown in Figure A, that combines head-mounted and projector-based displays to simultaneously provide a high-quality stereoscopic view of the procedure and a lower-quality monoscopic view for peripheral awareness.¹ We use a Kaiser ProView 30 stereo HMD, which does not have baffling material around the displays that would block the wearer's peripheral view of the real world, and four projectors to render view-dependent monoscopic imagery on Gatorboard and Styrofoam arranged to match the operating table and nearby walls.

We evaluated the display's efficacy with a formal human subject study. We asked the subjects to solve a series of word puzzles directly in front of them while simultaneously detecting and identifying objects entering and exiting the periphery.

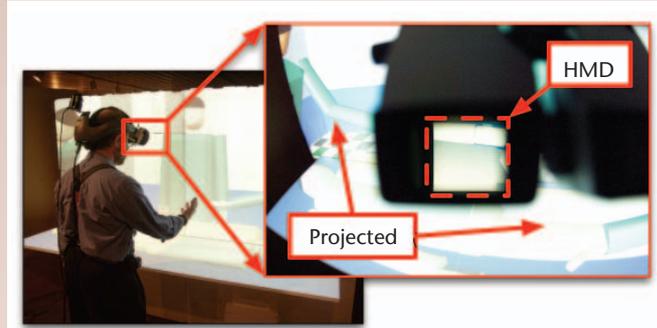


Figure A. Our hybrid display system combining head-mounted and projector-based displays.

Our study indicated that peripheral distraction and associated head turn rates were significantly less when solving both sets of tasks using the hybrid display than an HMD alone.¹

Reference

1. A. Ilie et al., "Combining Head-Mounted and Projector-Based Displays for Surgical Training," *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 2, Apr. 2004, pp. 128-145.

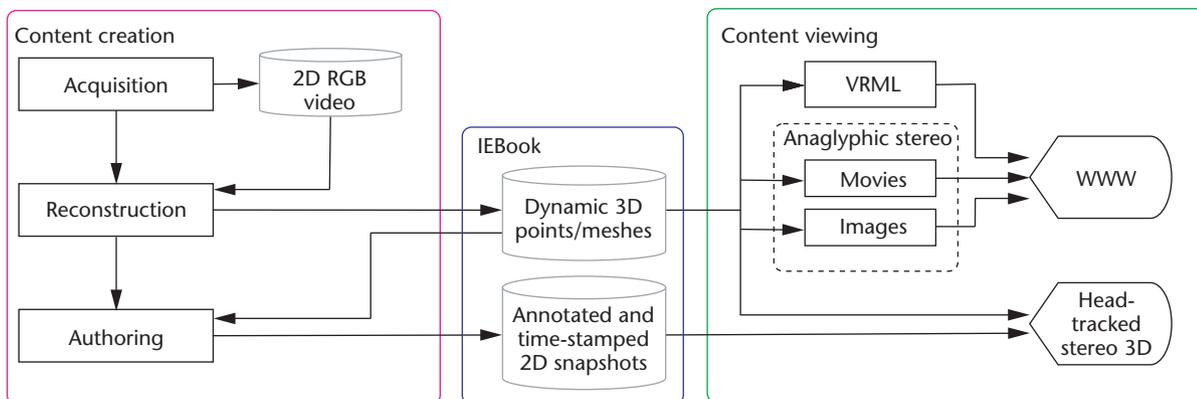


Figure 4. Overview of the phases involved in creating and viewing an immersive electronic book (IEBook). (RGB = Red, Green, Blue; VRML = Virtual Reality Modeling Language; and WWW = World Wide Web.)

<http://www.parallelgraphics.com>).

We generate anaglyphic (red-blue) stereo images and movies from the dynamic 3D points/meshes so users can view the data in stereo, using an inexpensive pair of red-blue 3D glasses. We select representative moments within each sequence and render stereoscopic still images of the points/meshes at those moments. We then make these available with the Java Stereoscope

applet (see <http://www.stereofoto.de>). We also create anaglyphic stereo movies in QuickTime and other formats.

Content creation

Figure 4 gives an overview of the two major phases involved in IEBook production: *content creation* and *content viewing*. During content creation, the system acquires 2D video of an actual

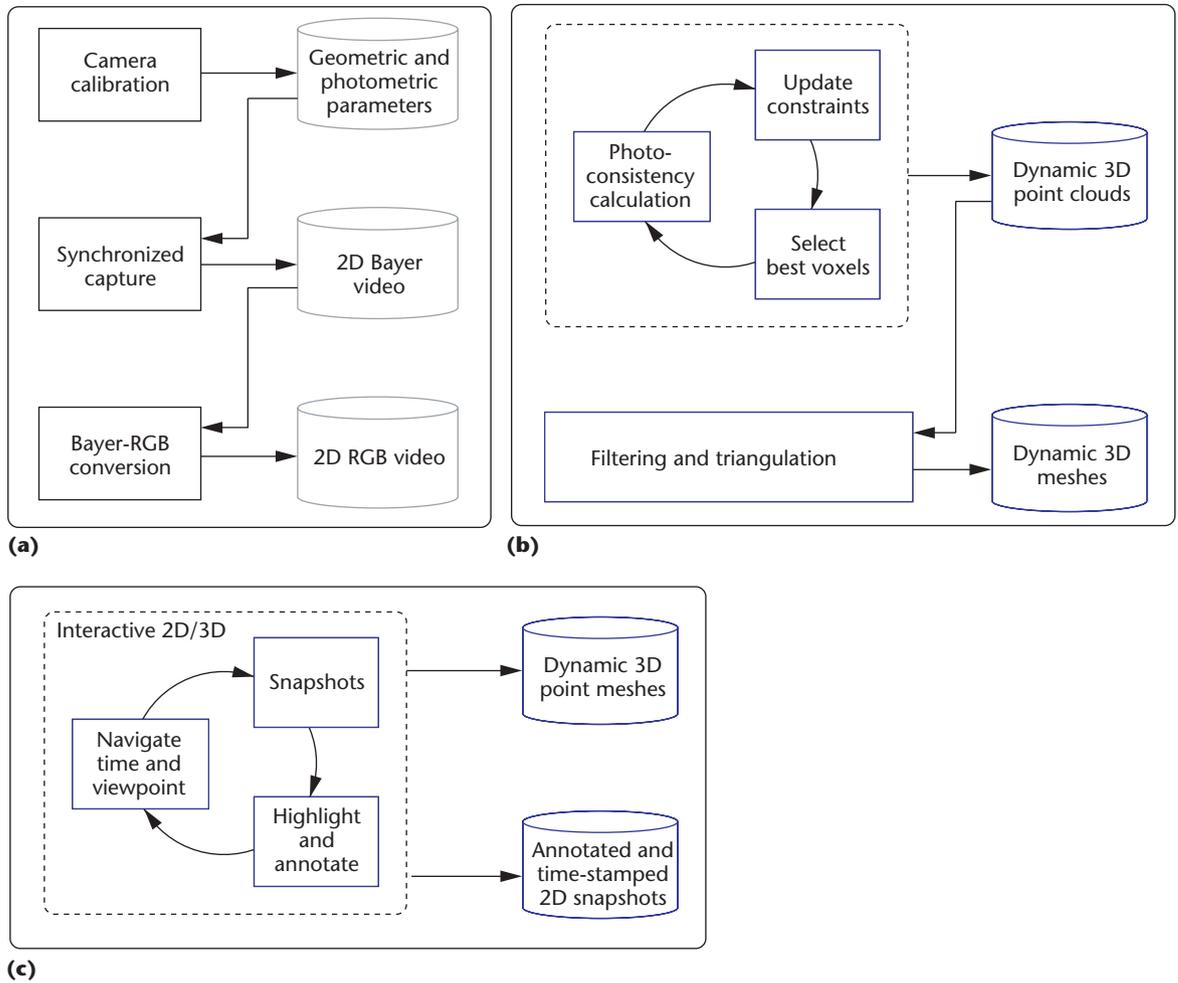


Figure 5. Content creation involves three steps: (a) acquisition of the 2D video of an actual surgical event, (b) reconstruction of 3D models, and (c) authoring an IEBook.

surgical event from multiple cameras, reconstructs 3D models, and authors an IEBook. Figure 5 illustrates these processes.

Acquisition

The first step in creating an IEBook is capturing (acquiring) the event of interest using the camera cube. The basic process, depicted in Figure 5a, involves calibrating the cameras, capturing synchronized video, and converting the raw images into the RGB color space for reconstruction.

Camera calibration. To perform the vision-based 3D reconstruction (see Figure 5b), we need estimates of analytical models for the cameras' geometric and photometric properties. We perform respective calibration procedures to prepare for event capture.

For geometric calibration, we use Bouguet's Camera Calibration Toolbox for Matlab.³ We take a sequence of frame sets (sets of eight synchronized images—one from each camera on the camera cube) of a moving black-and-white checked calibration pattern (see Figure 6), and then use the toolbox to estimate the intrinsic (focal length, image center, and pixel skew) and extrinsic (rotation and translation) parameters and lens distortion coefficients for each camera.

Because we use a color-based reconstruction approach, the color responses of the individual cameras must match. To achieve this, we use a closed-loop approach that seeks to adjust some hardware registers in the PGR cameras for optimal color matching and then compensates for remaining errors in software. By setting the hardware registers to their optimal values during acquisition, we in effect apply part of the photo-

metric calibration. Residual errors are corrected during a software refinement step.

To determine optimal values for the eight cameras' hardware registers, we developed software that iteratively adjusts the registers while observing the effects in images of a Gretag Macbeth "Color-Checker Color Rendition Chart" (see Figure 6). The software adjusts the parameters in each camera and color channel such that the resulting images are as alike as possible. Specifically, we use Powell's method for nonlinear optimization⁴ to minimize a cost function defined as the weighted sum of the squared differences between color values, and the variances across the sampling window of each color sample. The parameters of the software refinement step are computed in a onetime final calibration stage.

Synchronized capture. To improve the quality of the reconstruction results we set the camera cube on top of black fabric as shown in Figure 6. Before capturing an event, we acquire imagery to estimate the actual color of the fabric, then during reconstruction we mark as "background" any scene points that appear to have that color.

To capture an event, we first start the image capture server software on each of the four Dell servers, and start a master synchronization program that sets off the synchronized capture to disk. The capture continues for a preset number of frame sets.

We address server bandwidth and disk space concerns both by design (the camera-group configuration of Figure 3) and by removing old files and defragmenting disks prior to capture. Each 640×480 camera image is approximately 300 Kbytes. At a capture rate of 15 images per second, with two cameras per server, we need approximately 10 Mbytes per second of bandwidth from the cameras to the server disks, and approximately 0.5 Gbytes of disk space per minute of capture per server.

Color space conversion and distortion removal. As with most single-sensor cameras, the charge-coupled device (CCD) in the PGR cameras achieves color imagery using optical filters that spatially divide the 640×480 pixels (picture elements) among red, green, and blue (RGB). The CCD in our PGR cameras uses a spatial arrangement called the Bayer Tile Pattern. As such, each native captured 8-bit image contains interspersed data from all three color channels that a demosaicing algorithm combines to create

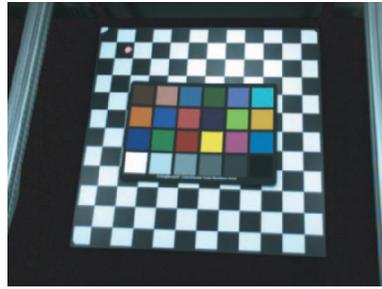


Figure 6. The geometric and photometric calibration patterns, as seen from one of the eight cameras. The patterns shown together here are actually used separately for geometric and photometric calibration.

a 640×480 24-bit RGB image. We developed software that uses PGR libraries to convert the native 8-bit Bayer images into 24-bit RGB images. Our code also uses the Intel Image Processing Library to remove the lens distortion estimated during the geometric calibration.

Reconstruction

The 3D reconstruction process involves two major steps:

- reconstruction of 3D points from 2D images using view-dependent pixel coloring (VDPC) and
- reconstruction of 3D surfaces from the 3D points using application-specific point filtering and triangulation to create 3D meshes.

Figure 5b gives an overview of the reconstruction process.

View-dependent pixel coloring. VDPC is a hybrid image-based and geometric approach that estimates the most likely color for every pixel of an image that would be seen from some desired viewpoint while simultaneously estimating a view-dependent 3D model (height field) of the underlying scene. (Further theory and implementation details of VDPC and a discussion of related work are available elsewhere.⁵) By considering a variety of factors, including object occlusions, surface geometry and materials, and lighting effects, VDPC produces results where other methods fail—that is, in the presence of textureless regions and specular highlights, which are common conditions in surgery (skin, organs, and bodily fluids).

Figure 7 illustrates VDPC's fundamental concepts. We begin by defining a 3D perspective voxel grid from the desired viewpoint, which is typically situated above and looking down into the camera cube, as Figure 7a demonstrates.

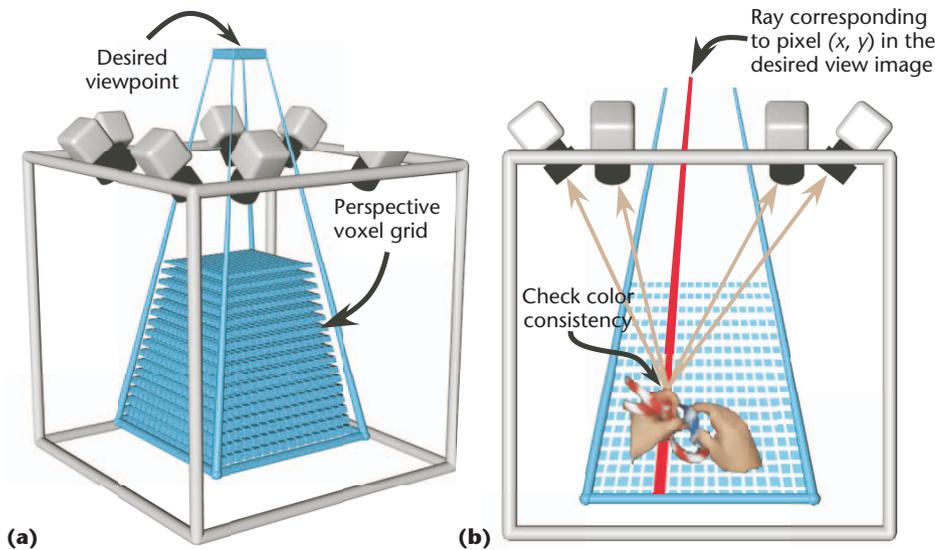


Figure 7. View-dependent pixel coloring (VDPC): (a) the desired viewpoint and perspective voxel grid concepts, and (b) the color consistency check for a particular voxel along a ray from the desired viewpoint image. The arrows extending from the hand toward the cameras depict the reflectance or imaging of the surface (voxel) into or by the cameras.

As Figure 7b illustrates, for each x - y pixel in the desired viewpoint image, we effectively traverse the ray away from the desired viewpoint to estimate the most likely color for that pixel. To do this, we test each voxel along the ray by back-projecting it into each of the eight cameras and looking at the actual camera image color at that point. We choose one “winner” voxel (the voxel with the most plausible back-projected appearance in all camera samples). We then use the median of the winner’s back-projected camera sample colors as the surface (voxel) color estimate, and the position on the ray as the 3D coordinate of the surface point. We mark all winner voxels as opaque surfaces, and all others along the ray between the desired view and the surface as empty. Like Kutulakos’s and Seitz’s seminal space-carving work,⁶ this effectively carves away voxels that do not appear to be part of a surface.

We repeat this volume-sweeping process, allowing estimated opaque surface points to occlude other voxels, progressively refining an estimate of a height field corresponding to the first visible surfaces. The dashed box in Figure 5b illustrates this cycle.

Two aspects of VDPC contribute to its relative robustness in the presence of textureless and specular surfaces.

First, we use a *view-dependent smoothness constraint*, assuming that 3D surfaces are more likely

to be continuous without abrupt change in depth. We implement the constraint with a new volumetric formulation of the relatively well-known disparity gradient principle. VDPC uses this constraint, updating the likelihoods as it progressively refines the surface voxel model.

Second, VDPC uses a *physically based consistency measure*. When sweeping along x - y desired viewpoint rays, we declare the voxel with the most plausible back-projected appearance the winner. One measure of plausibility is the variance of the back-projected color samples—the smaller the variance the more likely a voxel is to be a surface voxel. Our physically based consistency measure goes farther, allowing for back-projected sample color distributions that are plausible in the presence of specular highlights on many types of surfaces. In

these cases, the colors of back-projected samples occurring near specular reflections of a light source tend toward the light’s color. In fact, in the presence of these specular highlights, the eight back-projected sample colors will lie along a line in the RGB color space, extending from the surface’s inherent color toward the light’s color. By allowing for this specular distribution of back-projected sample colors, we can detect surfaces where simple variance tests would fail.

When we apply VDPC to a frame set, we get a static surface voxel model from the desired viewpoint’s perspective—that is, a height field. In fact, because some x - y rays finish without winner voxels and some are declared part of the background, the result is a sparse surface voxel model. We use the estimated camera geometry to transform this into a 3D point cloud in the camera cube space, and repeatedly apply the overall approach to all of the frame sets to obtain a dynamic 3D point cloud.

Filtering and triangulation. After reconstruction, some holes will exist in the 3D point data, and some 3D points will visually appear to be outliers. If we filter and triangulate the 3D points where we believe there are indeed surfaces, we can obtain more continuous dynamic models, further improving the appearance.

To this end, we developed new postprocessing methods to filter and triangulate the dynamic

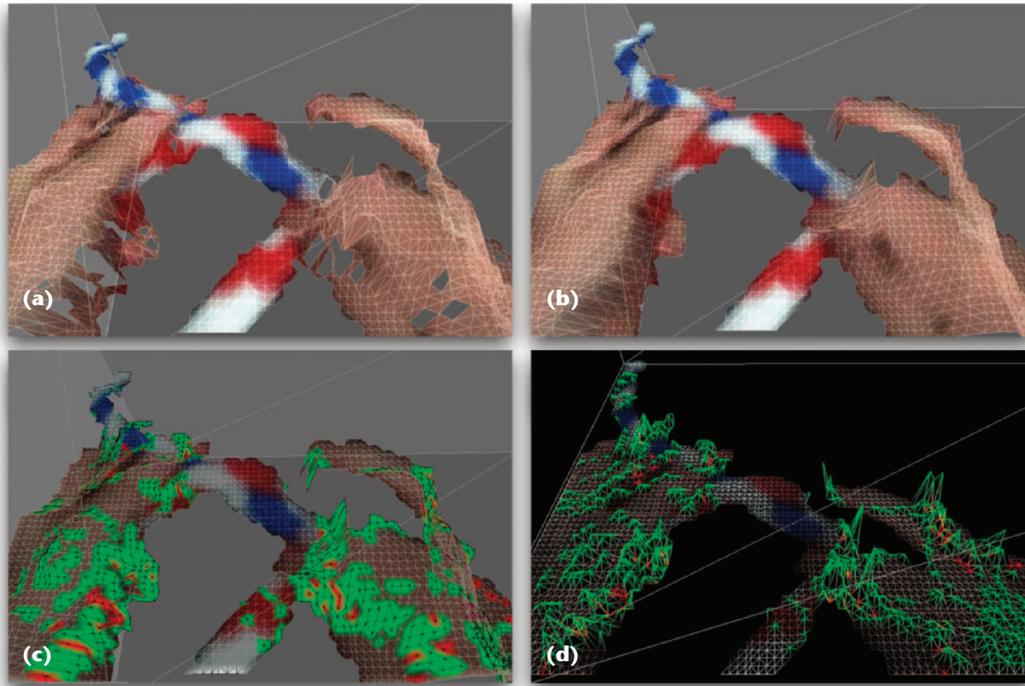
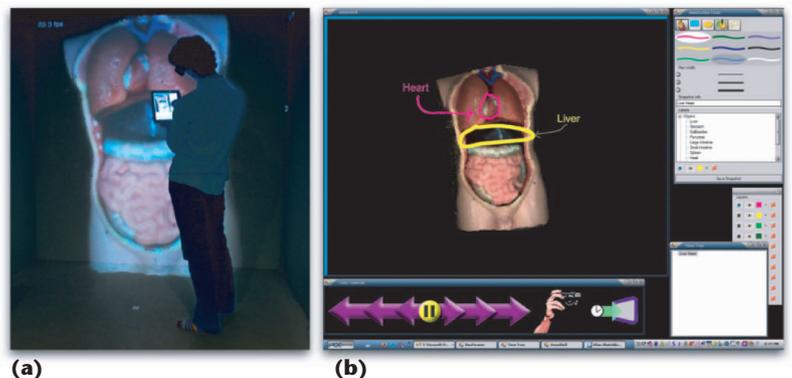


Figure 8. Filtering and triangulation: (a) triangulation without filtering, (b) triangulation with filtering, (c) filtered points colored in green and filled holes in red, and (d) absolute filter correction values as a height field. The highest spikes correspond to pointy outliers in (a).

sparse 3D point data. Figure 8 shows some results. Although we could use various off-the-shelf methods, VDPC allows for more specific constraints. For example, because the reconstructed points are on a 2D view-dependent grid, we can use 2D algorithms for filtering and triangulation. In addition, VDPC assigns a unique color to points estimated to be on a surface's edges, letting us "break" meshes at edges and helping us identify holes within the reconstructed shapes.

Our postprocessing filtering and triangulation software applies three steps in order:

1. *Median filtering.* The software uses a 5×5 median filter to smooth the z values. For each sample, the software inspects the 24 grid locations in the surrounding 5×5 x - y square. If at least three of these locations contain samples (including edge points), we replace the original sample's z value with the median value.
2. *Hole filling.* Exploiting the edge-color contouring, the software fills holes by examining neighboring samples. For each unpopulated sample located in a hole, the software uses a 3×3 box filter to average both color and depth values from up to eight neighboring samples.
3. *Triangulation.* Finally, the software examines the array in square groups of 2×2 samples. If any group is fully populated, we can use two



possible triangulation techniques. We pick the one that minimizes the color difference along the diagonal.

Authoring

Our current approach to authoring includes a novel combination of 2D and 3D interaction techniques. The primary motivation for a hybrid 2D/3D approach is to provide a familiar and tangible means of sketching (the notepad paradigm) while simultaneously offering a natural and immersive means of viewing the dynamic 3D data and the evolving IEBook.

Figure 5c illustrates the authoring process. Figure 9a shows an author annotating an IEBook in the VR-Cube, and Figure 9b is a screen shot of the authoring interface on the Tablet PC.

Using VCR-like time controls (see the right

Figure 9. Authoring an IEBook: (a) an author in the VR-Cube and (b) a screen shot of the interface on the Tablet PC.

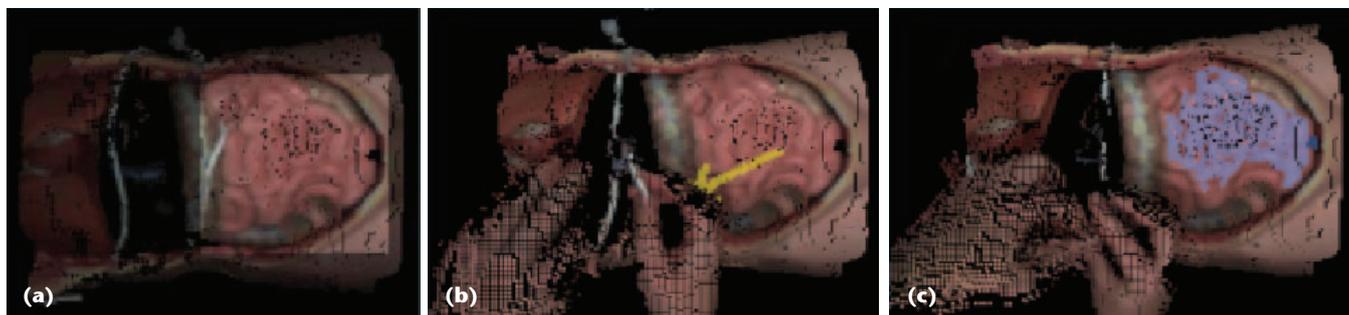


Figure 10. Examples of our dynamic 3D highlighting paradigms: (a) marquee, (b) freeform, and (c) fill. The tip of a freeform arrow in (b) is partially occluded by the surgeon's hands, illustrating the 3D nature of the highlights. Note that this data set has not been filtered and triangulated.

side of Figure 9b), the author navigates time in the captured sequence, looking for an interesting or important event. The author moves to a viewpoint where he or she has a good view of the surgeon's actions at that moment, and takes a snapshot of it using a button on the Tablet PC. Using the same Tablet PC interface, the author can highlight features, annotate the snapshot, and save the results to a virtual gallery in the IEBook. The author can arrange the snapshots hierarchically by dragging their titles on the Tablet PC application.

Unfortunately, conventional methods for selection and highlighting in 2D images do not accommodate depth or time. We therefore adapted some conventional methods to enable the selection and highlighting of dynamic 3D point/mesh data.⁷ Specifically, we implemented three distinct dynamic 3D highlighting paradigms: marquee, freeform, and fill, which Figure 10 illustrates. The implementations preserve highlights throughout a sequence and are adaptive, reacting to changes in the 3D topology over time.

To create a marquee highlight, the author drags a rectangle (or oval) around an area of interest in a 2D snapshot. The authoring application renders the selection into the current frame of the dynamic 3D data, darkening all points outside the selected area by an adjustable amount. The effect is akin to a square (or oval) spotlight illuminating the region of interest. The author can make as many selections as desired in the current frame. When the user presses play, the authoring software attempts to highlight the same regions in subsequent frames until the animation pauses. We also implemented a surface-specific variation that only highlights actual surfaces within the marquee region, omitting holes or background regions.

The freeform highlight is a paintbrush-style tool for creating colored marks on a fixed surface within the point cloud. To create a freeform highlight, the author clicks and drags the mouse to create a mark in the tablet display. When the author releases the mouse, the system renders the highlight onto the current frame as a tint of the selected color. When the author highlights subsequent frames, portions of the highlight are obscured whenever their corresponding surface points are obscured. The freeform highlight is surface specific in that the highlight's location is based entirely on the surface points' positions.

The fill highlight is similar to a "paint bucket" tool. It applies a tint of the current highlight color to a group of contiguous points around a selected location, all of which have similar depth and color values. Like the freeform highlight, however, the fill highlight can shrink and stretch, letting it compensate for small amounts of movement in the highlighted region.

The IEBook stores the highlight information as a series of 2D Boolean matrices indicating which vertices should be highlighted in each frame. Currently, we have 10 independent highlight layers, and can assign each to a different color. We can also hide the layers, letting the user see the results with or without their effects.

Content viewing

Our primary paradigm for experiencing an IEBook is head-tracked stereo 3D, in which a student uses the VR-Cube and a subset of the authoring interface. This IEBook environment combines immersive 3D imagery with Tablet PC-based navigation and a hierarchy of annotated snapshots.

A student views the snapshot gallery in the VR-Cube, holding the Tablet PC as in Figure 9.

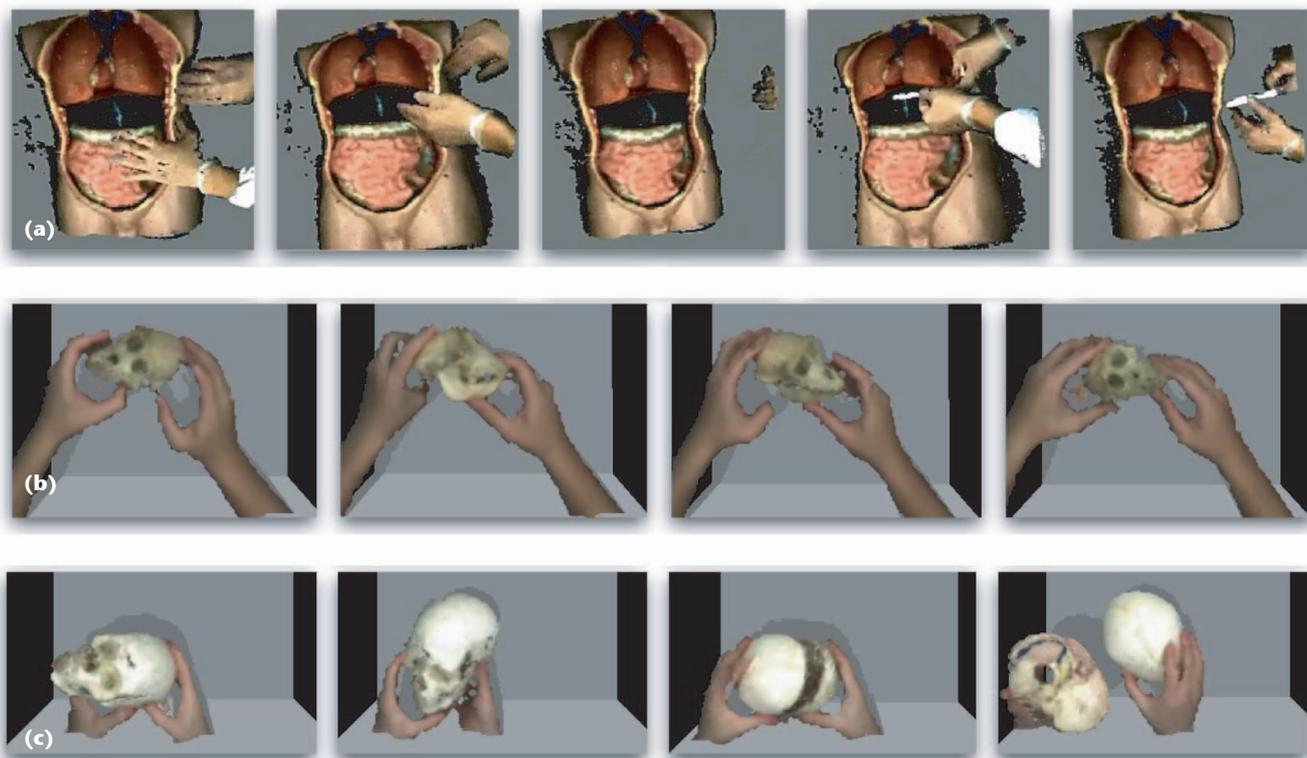


Figure 11. Reconstruction results: (a) a mock procedure to manage blunt liver trauma, (b) manipulating a Rhesus monkey skull, and (c) a human skull from a 12-year-old child. The human skull is a teaching artifact that can be separated into multiple parts. Both skull reconstruction sequences include simulated dynamic shadows.

The user can navigate to a specific point in time using the VCR-like controls or by choosing a snapshot in the hierarchy. A blue frame provides feedback about the active snapshot during the selection process. When the user selects a snapshot, the model jumps to the time associated with it.

We built the software for our hybrid display system prototype (see the “Visualizing Procedures Up Close and All Around” sidebar) on the same scene graph software used in the Barco TAN VR-Cube implementation. However, we added a rendering mode to allow rendering in both a head-mounted display and with projectors illuminating disjoint and casually arranged (not in a cube) planar surfaces. (We currently use this equipment to show only the dynamic 3D points/meshes from the reconstruction phase of content creation and not the results of authoring.)

Although we believe the head-tracked stereo 3D systems and VR-Cube provide the greatest sense of immersion, they are typically available only in visualization labs. So that more people can experience our system, we made it available

on the Web. In consultation with our medical partners, we chose three media: anaglyphic stereo images, anaglyphic movies, and nonstereo but dynamic (in space and time) VRML models. We developed software to generate anaglyphic stills and movies from the 3D point/mesh data and separate software that creates “boxed” VRML data sets. Each of the latter features a 3D VCR-like user interface to play and step through a sequence, including a slider bar for random access similar to typical computer-based movie players.

Technical results

Figure 11 shows several views of a dynamic reconstruction of Bruce Cairns performing a mock procedure to manage blunt liver trauma (from Figure 2b), and another coauthor manipulating Rhesus monkey and human skulls.

Figure 12 (next page) shows a preliminary IEBook in the VR-Cube. The six annotated snapshots correspond to six (mock) items of interest. The third item is selected and playing. Zoom boxes magnify the imagery in the screen capture

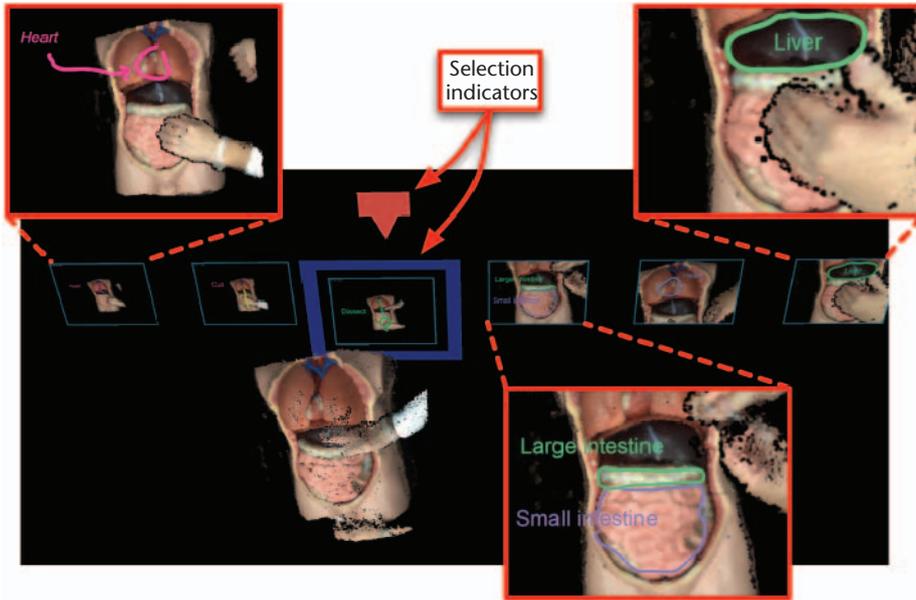


Figure 12. Screen shot from the VR-Cube showing several annotated snapshots arranged in an IEBook virtual gallery. Zoom boxes magnify three of the snapshots, and the blue frame and red marker indicate the current selection.

for several items. (The annotations are clear to an actual user, but do not reproduce well in a relatively small image.)

One of the skills most needed by new surgeons, yet difficult to learn, is suturing. Knot-tying skills in particular can be critical—the wrong knot can cause a wound to open or result in tissue damage. The surgeons we know agree that 2D knot-tying images and knot-tying movies are woefully inadequate. We therefore attempted to acquire and reconstruct some fundamental knot-tying demonstrations. We can reconstruct hands and moderately thick rope, and have done so for a few basic surgical knots as well as some common sailing knots. Figure 13 shows an anaglyphic stereo movie and the VRML interface of a ring knot tying reconstruction.

Results showing basic sailing knots and common surgical knots are available at <http://www.cs.unc.edu/~ootf/Projects/ebooks/reconstructions/>. We will continue to add reconstruction results as we obtain them.

User reactions

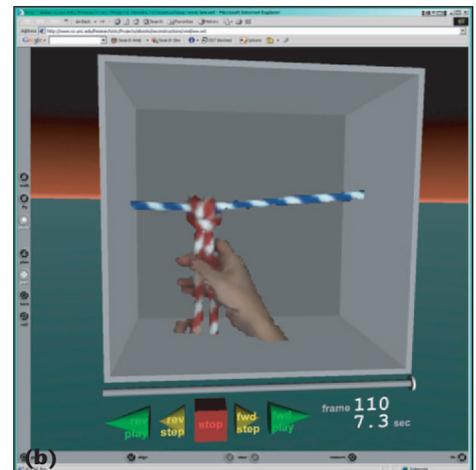
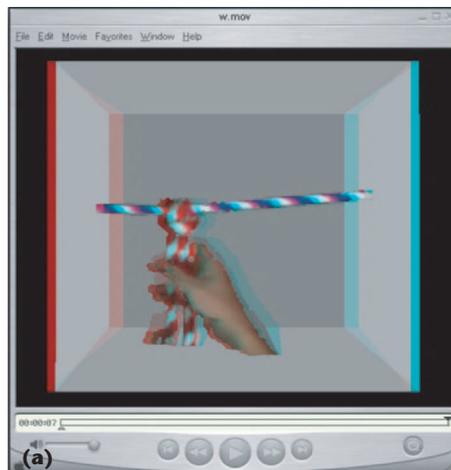
We have shown several surgeons our lab-based (stereo and head-tracked) and Web-based (movies and VRML) reconstructions (such as those in Figure 11, not the complete IEBook results in Figure 12). The surgeons' reactions have been very positive.

Most of the surgeons cited size, resolution, and visibility of the reconstructions as the primary limitations, which is no surprise to us at the project's relatively early stage.

We did not optimize the choice and arrangement of cameras in our camera cube (see Figure 2) for any particular procedure. Rather, we based our choices on what was practical for a proof-of-concept system supporting reconstructions of modest detail throughout a modest working volume.

As the “Visualizing Procedures Up Close and All Around” sidebar explains, both the procedure and the surrounding events are important. Achieving appropriate resolution in terms of reconstructions is a huge challenge. Ideally, we would have 3D reconstructions on the order of millimeters over the entire operating room, which might be 5–10 meters per side (125–1,000 cubic meters). Because this is currently impractical, near-term future challenges will include

Figure 13. Ring knot tying: (a) anaglyphic stereo movie (“red over right”), and (b) VRML interface. The movie includes simulated dynamic shadows; the VRML model does not.



choosing camera configurations and algorithms that can achieve high-resolution reconstructions where most critical, and relatively low-resolution reconstructions everywhere else.

Future work

Despite our achievements over several years, we are still in the early stages of a long-term effort. Many problems remain to be solved, and there is much work to be done to achieve the scale, fidelity, flexibility, and completeness that we envision.

In the near term, we will continue to reconstruct and process various objects and mock procedures. For example, we are procuring some rapid-prototype models of a real skull for a mock sagittal synostosis procedure. Soon we hope to provide anaglyphic stereo 3D data sets on the Web, letting viewers control the viewpoint and time. We are trying to solve problems with the network connection to the Tablet PC used for authoring and viewing. Although a wireless paradigm is clearly desirable, the VR-Cube hardware seems to interfere with the wireless connection.

A long-term primary goal is to scale our system up to enable the acquisition and reconstruction of (at least) a surgical table, the surrounding area, and the involved medical personnel. In fact, within several years we plan to equip an intensive care unit with numerous heterogeneous cameras to capture a variety of medical procedures. This involves solving nontrivial problems such as resolution, visibility, computational complexity, and massive model management. To assist with camera placement, we are working on a mathematical and graphical tool to help estimate and visualize acquisition information/uncertainty throughout the acquisition volume for a particular candidate set of cameras.

In light of our goals for fidelity and scale, we continue to improve our existing methods for 3D reconstruction and to investigate new methods. For example, we might combine pre-acquired laser scans of an operating room with camera-based dynamic reconstructions so we can better allocate cameras for dynamic events. We are also investigating possibilities for capturing real-time data from medical monitoring equipment, and plan to include it as metadata in an IEBook.

Whereas in the past we chose to hold off on audio acquisition, instead attempting to address the more difficult problems related to visual reconstruction, we plan to acquire audio as well.

We are also rethinking our choice of a Tablet PC

as the primary interface to the immersive authoring system. The tablet display is hard to read while wearing stereo glasses, and both hands are typically busy holding the tablet and the stylus, so removing the stereo glasses when necessary is not an attractive solution. In our current paradigm, authors choose a view with their heads (looking at the data of interest) while trying to use the tablet stylus to initiate snapshots, which is awkward. One possibility is to track the Tablet PC so users can choose snapshots in a "viewfinder" mode.⁸

Finally, we hope to increase the impact on the medical community by making complete IEBooks available on the Web. The primary difficulty here is in determining which interaction techniques are appropriate and how to implement them. Rather than simply "dumbing down" the fully immersive interfaces, we want to use the best interfaces for each paradigm and authoring tools that appropriately target each. **MM**

Acknowledgments

At the University of North Carolina, we acknowledge Marc Pollefeys for VDPC collaboration; Jim Mahaney and John Thomas for technical support; and surgeons Ramon Ruiz and Anthony Meyer for general collaboration. At Brown University, we thank Melih Betim and Mark Oribello for their systems and video support.

This research was primarily supported by US National Science Foundation Information Technology Research grant IIS0121657, and in part by US National Library of Medicine contract N01LM33514 and NSF Research Infrastructure grant EIA0303590.

References

1. H. Fuchs et al., "Virtual Space Teleconferencing Using a Sea of Cameras," *Proc. 1st Int'l Symp. Medical Robotics and Computer Assisted Surgery*, Shadyside Hospital, Pittsburgh, 1994, pp. 161-167.
2. A. van Dam et al., "Immersive Electronic Books for Teaching Surgical Procedures," *Telecomm., Teleimmersion, and Telexistence*, S. Tachi, ed., Ohmsha, 2002, pp. 99-132.
3. J.Y. Bouguet, "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
4. W. Press et al., *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, 1993.
5. R. Yang, *View-Dependent Pixel Coloring—A Physically Based Approach for 2D View Synthesis*, doctoral dissertation, Univ. of North Carolina at Chapel Hill, 2003.

6. K. Kutulakos and S. Seitz, "A Theory of Shape by Space Carving," *Int'l J. Computer Vision*, vol. 38, no. 3, 2000, pp. 199-218.
7. D.M. Russo, "Real-Time Highlighting Techniques for Point Cloud Animations," master's thesis, Brown Univ., 2004.
8. M. Tsang et al., "Boom Chameleon: Simultaneous Capture of 3D Viewpoint, Voice, and Gesture Annotations on a Spatially Aware Display," *Proc. 15th Ann. ACM Symp. User Interface Software and Technology*, ACM Press, 2002, pp. 111-120.



Greg Welch is a research associate professor of computer science at the University of North Carolina (UNC) at Chapel Hill. His research interests include tracking and sensing systems for virtual environments and 3D telepresence. Welch has a PhD in computer science from UNC-Chapel Hill. He is a member of the IEEE Computer Society and the ACM.



Andrei State is a senior research scientist in the Department of Computer Science at UNC-Chapel Hill. His technical interests include 3D graphics and mixed reality. State has a Dipl.-Ing. degree from the University of Stuttgart and a master's degree in computer science from UNC-Chapel Hill.



Adrian Ilie is a graduate research assistant in the Department of Computer Science at UNC-Chapel Hill. His research interests include camera placement for 3D reconstruction, photometric calibration, and image processing. Ilie has an MS in computer science from UNC-Chapel Hill.



Kok-Lim Low is a doctoral candidate in the Department of Computer Science at UNC-Chapel Hill. His research interests include projector-based rendering, augmented reality, and 3D imaging and modeling using active range sensing. Low has an MS in computer science from UNC-Chapel Hill.



Anselmo Lastra is an associate professor of computer science at UNC-Chapel Hill. His research interests are computer graphics, specifically image-based modeling and rendering, and graphics hardware architectures. Lastra has a PhD in computer science from Duke University.



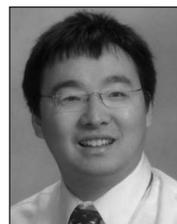
Bruce Cairns is an assistant professor at UNC-Chapel Hill and director of the North Carolina Jaycee Burn Center. His research interests include trauma, burns, critical care, cellular immunology, and shock. Cairns has an MD from the University of Pennsylvania.



Herman Towles is a senior research associate in the Department of Computer Science at UNC-Chapel Hill. His research interests include graphics hardware architecture, large-format projective displays, 3D scene reconstruction, and telepresence. Towles has an MS in electrical engineering from the University of Memphis.



Henry Fuchs is the Federico Gil Professor of Computer Science, adjunct professor of biomedical engineering, and adjunct professor of radiation oncology at UNC-Chapel Hill. His research interests include computer graphics and vision and their application to augmented reality, telepresence, medicine, and the office of the future. Fuchs has a PhD in computer science from the University of Utah. He is a member of the National Academy of Engineering and a Fellow of the ACM and of the American Academy of Arts and Sciences.



Ruigang Yang is an assistant professor in the Computer Science Department at the University of Kentucky. His research interests include computer graphics, computer vision, and multimedia.

Yang has a PhD in computer science from UNC-Chapel Hill. He is a recipient of the NSF Career Award, and is a member of the IEEE and the ACM.



Sascha Becker is a research scientist for the Brown Computer Graphics Group and a senior software engineer at Laszlo Systems in San Mateo, California. Her work focuses on using technology as a tool for art, education, and play. Becker has a BA in computer science from Brown University. More information is available at <http://sbshine.net>.



Daniel Russo is a software developer at Filangy, a start-up intent on revolutionizing Web searching. Russo has an ScM in computer science from Brown University.



Jesse Funaro recently earned an ScM in computer science from Brown University, with research in optimization techniques for search result ranking.



Andries van Dam is the vice president for research and professor of computer science at Brown University. His principal research interests are computer graphics, computer-based education, and hypertext and electronic books. Van Dam has a PhD in electrical engineering from the University of Pennsylvania. He is a member of the National Academy of Engineering and a Fellow of the ACM and the IEEE.

Readers may contact Greg Welch at the University of North Carolina at Chapel Hill, Department of Computer Science, CB# 3175, Sitterson Hall, Chapel Hill, NC 27510-3175; welch@cs.unc.edu.

GET CERTIFIED

Apply now for the 1 April—30 June test window.



CERTIFIED SOFTWARE DEVELOPMENT PROFESSIONAL PROGRAM

Doing Software Right

- Demonstrate your level of ability in relation to your peers
- Measure your professional knowledge and competence

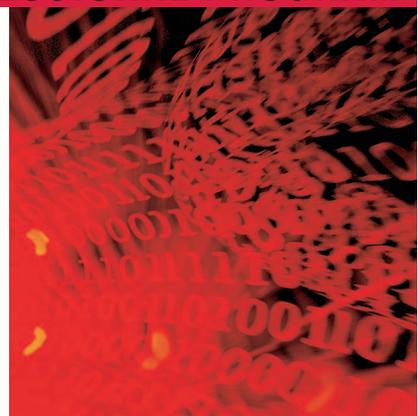
Certification through the CSDP Program differentiates between you and other software developers. Although the field offers many kinds of credentials, the CSDP is the only one developed in close collaboration with software engineering professionals.

"The exam is valuable to me for two reasons:

One, it validates my knowledge in various areas of expertise within the software field, without regard to specific knowledge of tools or commercial products...

Two, my participation, along with others, in the exam and in continuing education sends a message that software development is a professional pursuit requiring advanced education and/or experience, and all the other requirements the IEEE Computer Society has established. I also believe in living by the Software Engineering code of ethics endorsed by the Computer Society. All of this will help to improve the overall quality of the products and services we provide to our customers..."

— Karen Thurston, Base Two Solutions



Visit the CSDP web site at www.computer.org/certification
or contact certification@computer.org



July–September 2005