An Interactive Camera Placement and Visibility Simulator for Image-Based VR Applications

Andrei State[°•], Greg Welch[°], and Adrian Ilie[°]
[°]Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA 27599
[•]InnerOptic Technology Inc., Hillsborough, NC 27278

ABSTRACT

We describe an interactive software simulator that assists with the design of multi-camera setups for applications such as image-based virtual reality, three-dimensional reconstruction from still or video imagery, surveillance, etc. Instead of automating the camera placement process, our goal is to assist a user by means of a simulator that supports interactive placement and manipulation of multiple cameras within a pre-modeled three-dimensional environment. It provides a real-time 3D rendering of the environment, depicting the exact coverage of each camera (including indications of occluded and overlap regions) and the effective spatial resolution on the surfaces. The simulator can also indicate the dynamic coverage of pan-tilt-zoom cameras using "traces" to highlight areas that are reachable within a user-selectable interval. We describe the simulator, its underlying "engine" and its interface, and we show an example multi-camera setup for remote 3D medical consultation, including preliminary 3D reconstruction results.

Keywords: Camera placement, computer vision, coverage, geometry, interactive simulation, occlusion, reconstruction, surveillance, visibility.

1. INTRODUCTION

Many computer vision and virtual reality (VR) type applications use multi-camera setups with the aim of calculating three-dimensional (3D) reconstructions of static scenes or dynamic events. Surveillance-type applications make use of multiple cameras to ensure unobstructed coverage during observation of target events and environments. Choosing the number of cameras as well as their locations, orientations, fields of view and other parameters for such setups is a difficult task. Given a physical setting, a target located there and/or an event (possibly involving human activity) to take place there, how does one create an imaging setup that is suitable for the application at hand? For 3D reconstruction, one generally wants to ensure that the observed scene is completely imaged, such that a minimum number of camera views (often more than two) is available for every "reconstructible" location[3]. For surveillance, the visibility criteria may include camera placement close to eye level and guaranteed coverage of specific areas, such as entrances or exits. Surveillance applications may prefer complete coverage of critical areas to overlapping coverage, while 3D reconstruction may favor overlap over completeness. Other reconstruction and visualization techniques, such as image-based visual hulls[8], also have very specific requirements with respect to the geometric layout of their imaging cameras. This also applies to mixed-reality applications which allow interaction between real and virtual objects (or humans) and make use of video cameras to capture real imagery that they incorporate into an otherwise synthetic environment (e.g., work by Lok[6]).

Starting in the mid-1990s, research groups at the University of North Carolina have been working on a number of projects involving reconstruction of dynamic objects as well as people, from simultaneously acquired video streams, for applications such as tele-presence[5]. Recently, the "Immersive Electronic Books" (IEBooks) project[14] was aimed at capturing time-varying 3D recordings of surgical procedures, in order to provide an immersive VR visualization of a previously recorded procedure, for the benefit of medicine students as well as physician or surgeon trainees. Currently, the "3D Medical Consultation" (3DMC) project[15] aims to create similar visualizations, but in real-time, which we postulate would benefit patients in situations such as trauma surgery, where a non-specialist on-site trauma helper might receive valuable real-time help from remotely located specialists viewing the trauma patient and scene through a live three-dimensional link (Fig. 1).

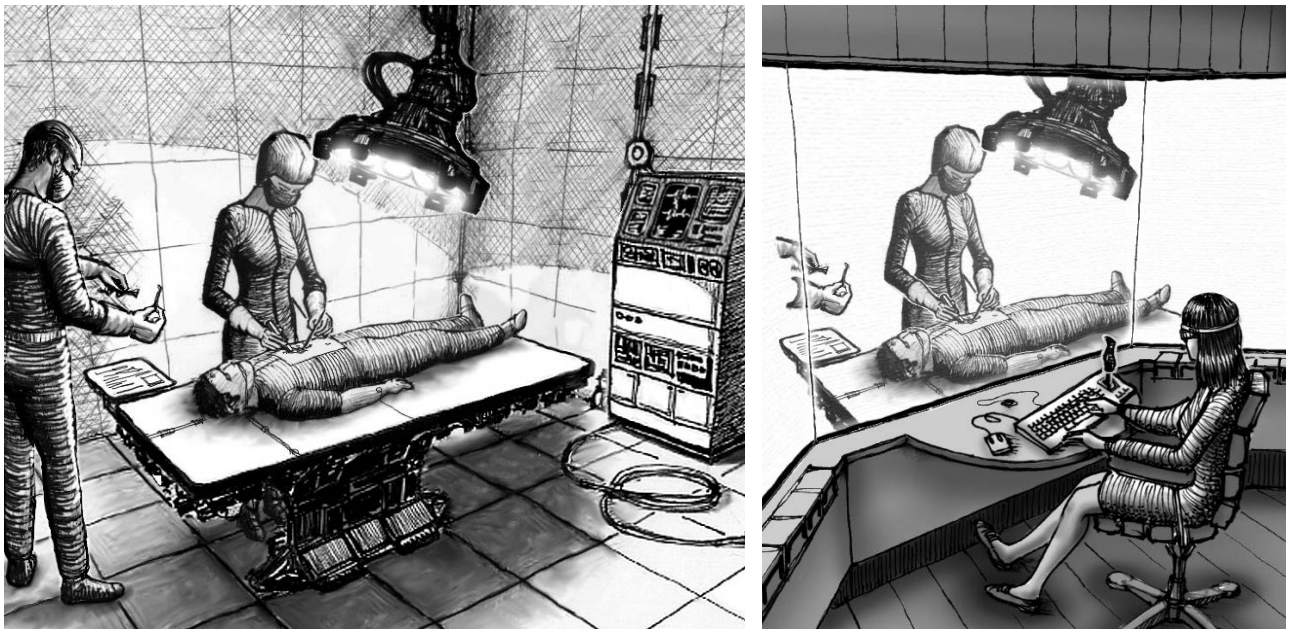


Fig. 1. Conceptual sketches of future 3D medical consultation scenario. Left: emergency room with trauma victim, on-site medical personnel and real-time 3D capture device (incorporated into the lighting fixture). Right: off-site trauma specialist consults while viewing real-time reconstruction of procedure via live stereoscopic, head-tracked visualization.

Both of the above projects use multi-camera setups for video stream capture. The IEBooks setup and software were inherited from a previous reconstruction application[18]. At the beginning of the 3DMC project, we became aware for the first time of the aforementioned difficulty in devising adequate capture setups, especially given the additional constraints we had to satisfy; in particular, we had to design and build a transportable multi-camera capture device capable of performing inside an operating room during a critical, possibly life-threatening medical situation. Moreover, our capture system would have to be operated close enough to the medical personnel and the patient to acquire imagery that is useful for remote consultation, but without interfering in any way.

While there has been some work on automatic optimization of camera placement (see Section 2), the approaches appear to be practically limited to a relatively small number of cameras in a limited configuration. In any case, we were uncomfortable handing off the design to an automated system, which might have produced results for which we had no intuition. We wanted to be able to see and assess the tradeoffs ourselves. As such we decided to develop a geometric simulation of the emergency room, the patient, and the medical personnel treating the injury. Furthermore, our simulation should display multiple camera "candidates" and the amount of scene coverage provided by each. We were initially inspired by the "pre-visualization" systems (such as StoryViz[13]) that are now commonplace in the motion picture industry and serve to simulate sets, actors, props, lights, effects and cameras, including their movements or other dynamic behavior. Such systems are used to optimize set design and choreography (for actors as well as cameras), and they are also used to create storyboards. The output from a pre-viz system is a stream of images that approximate the final motion picture imagery (for a fraction of the cost). Our problem is related but not identical: since we ultimately aim to obtain not a two-dimensional image of a scene but a three-dimensional reconstruction, we realized that we should not output simulated camera images but *project* a camera's coverage *into* the scene. We therefore propose to treat each camera as a projector illuminating the parts of the scene that it can see. With this in mind, a simple pinhole camera with a rectangular imager will be represented as a projector "emitting" into the scene an infinite light pyramid with a rectangular base. The objects that occlude other objects—as seen from the camera—will cast shadows onto them, which also matches the behavior of projected light.

In the following, we summarize previous work we are aware of, we then introduce our visibility simulator and describe its implementation. We follow with a real-world example: the 3DMC capture device mentioned above. We also show how we extended the simulator to pan-tilt-zoom cameras and end with concluding remarks and suggestions for future work.

2. PREVIOUS WORK

Luhmann[7] describes an early software system that computes and graphically displays 2D and 3D plots of selected imaging parameters as functions of other imaging parameters. The program understands a vast array of characteristics describing cameras, scene and imaged objects; the user inputs numeric data for sensors, lenses, digitizers, objects to be imaged (geometry, surface characteristics), camera intrinsics, pose, lighting, modulation transfer functions, as well as user-defined parameters and formulas that establish relationships with the program's built-in models. Aside from 2D and 3D plots, the program can also generate simple elevation and perspective wire frame views of the imaging setup under consideration.

Oksanen[10] describes a "measurement model design tool" for multi-camera video digitizing configurations aimed primarily at quality control, e.g., on an assembly line. It uses the computer-aided design tool AutoCAD™, with CAD models of the target objects, to simulate camera images that include representations of estimated measurement precision at selected points on the objects.

Chen[2] includes a chapter on "Determining Optimal Camera Configurations." The author describes an automated approach to the camera placement problem aimed at motion capture systems. While the approach is arguably restricted to a relatively small number of cameras in limited configurations, Chen presents a quality metric that accounts for both the camera resolution and likely occlusion characteristics of a camera configuration. It is this quality metric (with a probabilistic occlusion model) that Chen optimizes to arrive at a camera configuration.

Concurrently with the work described in this paper, co-author Greg Welch and collaborator B. Danette Allen have been working on a complementary method for comparing the expected performance of sensor systems[1]. They call their prototype *Artemis.* Their approach is to estimate the stochastic asymptotic (steady-state) position and/or orientation uncertainty at many points throughout the desired working volume, and then to visualize the results graphically. This global performance estimate is intended to provide both a quantitative assessment of the expected performance, and intuition about the type and arrangement of sources and sensors, in the context of the desired working volume, the expected spatial-temporal signal and noise characteristics of the sensors, and the expected scene dynamics. The approach is general in that it can be applied to virtually any sensor (magnetic, inertial, optical, etc.), including cameras. Applying such asymptotic analysis to our 3D reconstruction problem would yield a volumetric or surface dataset of reconstruction uncertainty values, which could be displayed with well-known visualization methods. Allen[1] shows the results of analyzing the IEBooks camera configuration mentioned above with Artemis. While we hope to integrate stochastic asymptotic position and/or orientation uncertainty visualizations into our system as an optional assessment tool, at present the related stochastic algorithms are too complex for interactive operation on contemporary hardware. This is true even if we restrict the computation to pre-determined areas such as the surfaces of pre-modeled scene geometry. The lack of interactivity in Artemis (resulting from the computational complexity) was part of the motivation for exploring the complementary geometric approach described in this paper.

3. OVERVIEW AND BASIC OPERATION

In contrast to Artemis, our real-time visibility simulator—which we call *Pandora*—is a purely geometric tool which takes into account camera parameters such as pose, field of view, resolution and lens distortion, and calculates and displays occlusion and visibility for a given set of geometric models. We built Pandora as an interactive 3D program with a graphical user interface (GUI) and a six-degree-of-freedom double joystick controller for viewpoint control and for manipulating the simulated cameras. Pandora shows real-time perspective views of the pre-modeled scene, together with the simulated cameras and their coverage. There is a main viewport, rendered from a user-controlled viewpoint; separate viewports rendered from the point of view of each simulated camera can also be displayed (Fig. 2, left).

At startup, the program loads a static, pre-modeled scene (consisting of polygonal objects) for which we aim to prepare a multi-camera imaging setup. The pre-modeled scene represents an approximation of the real-world environment to be captured. The multi-camera setup being worked on is loaded from a configuration file. Each camera is represented by a small cylinder and a colored light pyramid. In order to distinguish between the cameras, we assign a different color to the light emitted by each camera.

The models can be displayed as polygonal objects, using their intrinsic colors and textures (Fig. 2 left). In order to improve visibility in the areas illuminated by the cameras' light pyramids, we can also display the models as black polygons with grey hidden line edges (Fig. 2, center), or simply as black shapes with grey hidden line contours (Fig. 2, right). Then the only remaining colored surface areas will the parts of the scene illuminated by the light pyramids, i.e. visible to the simulated cameras. Moreover, areas that show superimposed light originating from more than one camera indicate overlapping coverage, such as the right eye in Fig. 2. Conversely, black areas are not illuminated and are therefore not visible from any camera, such as the left cheek in Fig. 2.

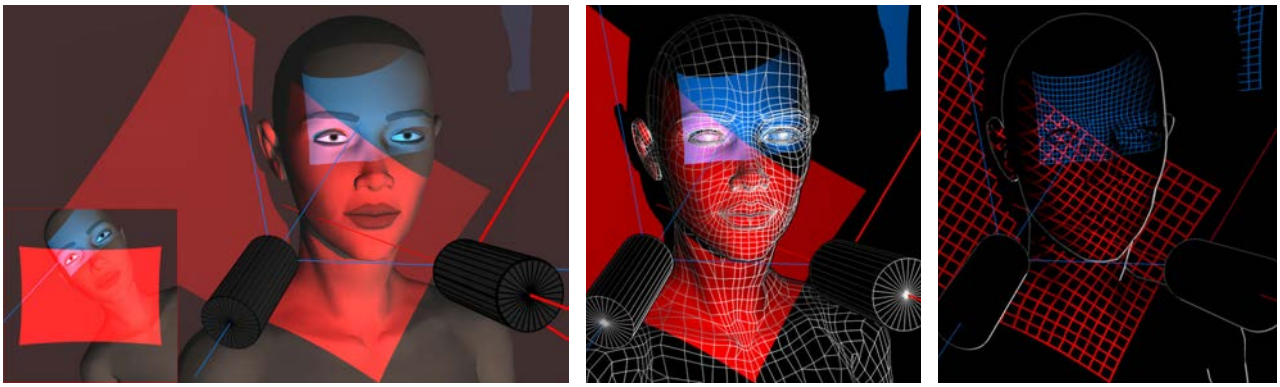


Fig. 2. Pandora display with pre-modeled scene and two simulated cameras. Left: models shown with intrinsic colors. The small viewport at the bottom left is the view from the currently selected camera (the one projecting the larger, red light pattern). Center: hidden line edge display. Right: hidden line contour display; the grid textures indicate camera resolution on the target surface.

The projected light pyramids can be uniformly bright or can be modulated with a 2D spatial pattern. The pattern can be structured to indicate local imaging resolution at each imaged point in the scene. For example, a simple grid pattern (Fig. 2, right) can be used to "transfer" (i.e. project) information about the imaging resolution (i.e., the pixel density in a camera's CCD array) onto the surface being imaged. Thus the user can asses whether a camera can resolve a specific feature on an object (e.g., facial features of a person passing through a surveyed doorway), which obviously depends on numerous factors such as camera-to-object distance, camera resolution and field of view, feature size, surface orientation with respect to the camera, etc. For example, in Fig. 2 (right) the blue-light camera images the right eye at much higher resolution than the red-light camera, since its projected grid pattern appears much denser.

In order to prepare and optimize the camera arrangement based on the coverage requirements dictated by the application, the user can navigate through the modeled scene to closely inspect areas of interest while interactively posing and zooming each simulated camera. This is accomplished by switching the double joystick controller between the main viewport and each of the cameras. (The readouts from the joystick's channels are floating-point numbers; we use a nonlinear fourth power mapping to achieve both rapid translation and fine movement for close-up inspection of small geometric features.) Pandora's GUI provides a number of additional controls for each camera: on/off, aspect ratio, field of view and lens distortion parameters.

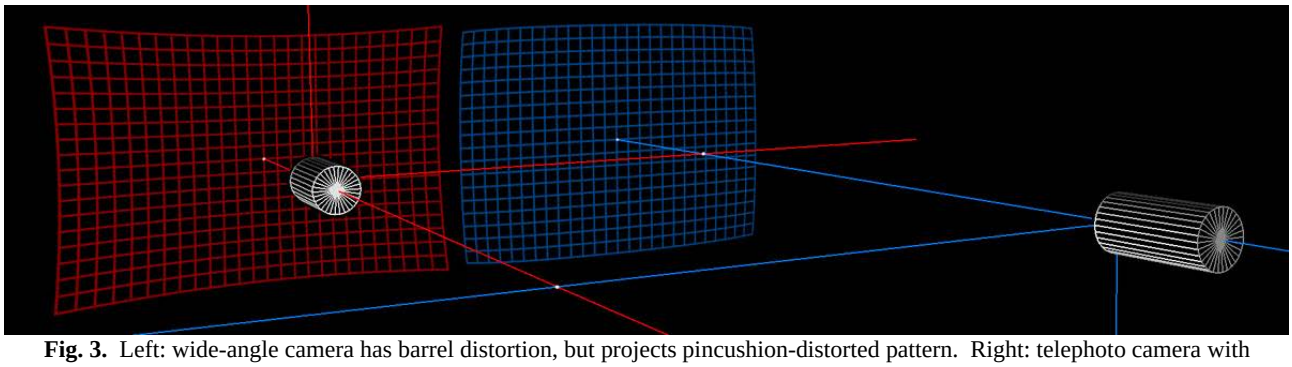


Fig. 3. Left: wide-angle camera has barrel distortion, but projects pincushion-distorted pattern. Right: telephoto camera with pincushion distortion projects barrel-distorted grid. See text for explanation.

For cameras with distortion-free lenses, we project an infinite light pyramid with a rectangular base, as already mentioned. The aspect ratio of the rectangular base matches the aspect ratio of the camera (and the pyramid's angles match the camera's horizontal and vertical fields of view, respectively). For the lens with optical distortion, we must warp the cross-section of the pyramid to accurately represent the area of space that can be imaged by such a lens. Typically, wide-angle lenses (or zoom lenses at their wide-angle setting) exhibit barrel distortion. Telephoto lenses (or zoom lenses at their telephoto setting) are characterized by small amounts of pincushion distortion. Due to our projection metaphor, we must "invert" the distortion: hence a barrel-distorted camera image is represented as a pincushion-distorted light pyramid, and vice-versa. The reason for this is that a camera with a barrel-distorting wide-angle lens can "see more" in the corners of its image, therefore Pandora must project a light pyramid with a pincushion-distorted base to simulate such a camera (Fig. 3).

4. IMPLEMENTATION DETAILS

Pandora was developed in C and C++ and uses the OpenGL API. To achieve interactive performance, it uses the following well-known rendering techniques that are supported on most modern graphics accelerators:

Projective textures. The colored light pyramids are implemented via OpenGL's projective texturing mechanism. (For each simulated camera, the projection origin corresponds to the center of projection (nodal point) of the camera's lens.) The projected pattern can be an arbitrary texture image. We use fixed-size 1024 × 1024 projective textures containing either uniform color fields or grid patterns, as described above. To allow interactive distortion of the projected patterns, we pre-render them (only when the user changes distortion parameters from the GUI) as two-dimensional images, then transfer them into texture memory.

Shadow mapping and multitexturing. Shadow mapping[17] operates by rendering a depth buffer of the scene from the point of view of the light projector. In hardware-assisted operation, the depth buffer is copied into the *shadow map* texture. During the main rendering pass, the distance of each rasterized fragment from the origin of projection is compared with the shadow map depth at the same location. Fragments that are deeper than the shadow map are either located on the back surfaces of objects or are occluded from the projection origin by other objects. The projective texture must not be applied to these fragments. To that end, we make use of the multitexture capability available in modern graphics accelerators. During the rendering pass that illuminates the scene with a specific projective texture, we also enable the shadow map texture (as well as a point light source also located at the origin of projection).

Note that a texture projector's shadow map need only be updated whenever the corresponding camera's position, orientation, field of view or distortion parameters change (the shadow map is not updated when the viewpoint or field of view of the main viewport change as the user navigates the scene). We use a shadow map resolution identical to the projective texture resolution (1024 × 1024). In order to maintain maximum accuracy in the shadow map, the depth bounds of the shadow map are kept as "tight" as possible. This is accomplished by pre-rendering a temporary low-resolution (currently 256 × 256 pixels) depth buffer—using generous tentative values for near and far clipping planes. The buffer is then searched for the largest and smallest value, which are subsequently used as the near and far clipping planes of the shadow map texture.

Blending. As mentioned, we use a different color for each simulated camera's projected texture. The parts of the scene that are visible from several cameras will have multiple overlapping patterns projected onto them. In order to be able to assess per-camera visibility and resolution conditions in such areas, we use additive blending of the projected textures. Thus, surface areas visible from both a red-light camera and a blue-light camera will appear colored in magenta (Fig. 1, center). On surface areas that are visible from many cameras, the accumulated projected color imagery saturates to white and it is no longer possible to distinguish between the various coverage areas. We use a global brightness slider (which affects all projected textures and hence also their blended sum) to alleviate this problem (Fig. 4).

Hidden line and contour rendering. For each active camera (or texture projector) in the scene, a separate rendering pass is required (with 2 texture units simultaneously enabled per pass as mentioned above). After these projective passes, the depth buffer is fully populated with all front-facing geometry, and the color buffer shows all projected (and possibly superimposed and blended) texture patterns, while the remaining geometry is black. In order to render the complete hidden line edges (Fig. 2, center), we render the front-facing geometry as lit wireframes. In order to render

the hidden line contours (Fig. 2, right), we render the back-facing geometry as lit wireframes. Since the depth buffer already contains the depth of the front-facing geometry, only the outlines of unoccluded objects remain visible[11].

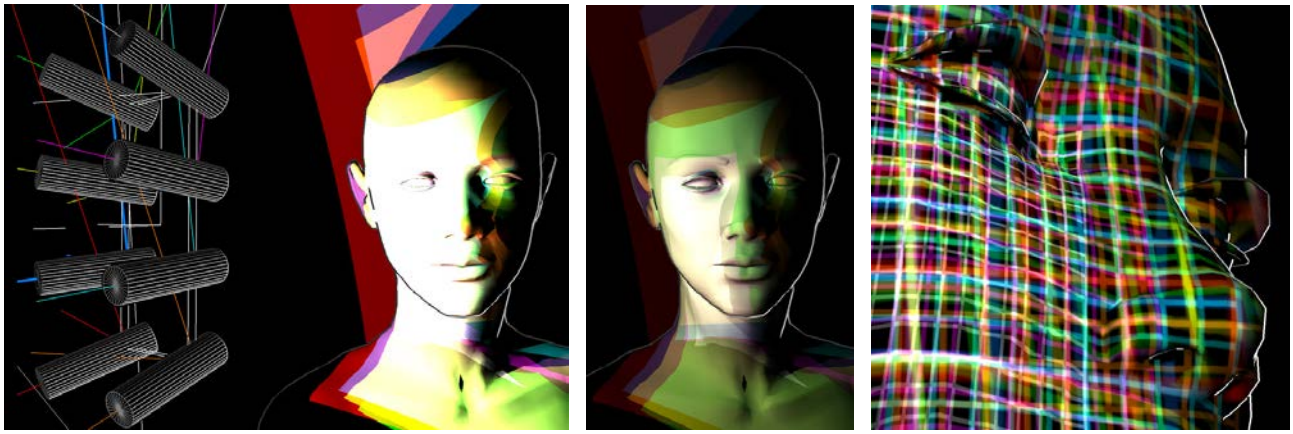


Fig. 4. Overlapping coverage with numerous cameras. Left: multiple projective textures clamp to white, and it is impossible to distinguish individual contributions. Center: reducing the brightness of all projectors alleviates the problem. Right: in grid mode the display becomes more cluttered, but one can move closer to asses imaging resolution on small areas.

On a modest personal computer (1.5GHz, 512MB main memory) equipped with a standard graphics accelerator (NVIDIA 6600GT, 128MB), Pandora runs at real-time frame rates (10-30Hz) with scene models consisting of tens of thousands of polygons and 8 simulated cameras. When all simulated cameras are grouped and moved at once, the frame rate slows down dramatically to around 1Hz, since the system must recalculate all eight shadow maps for each frame.

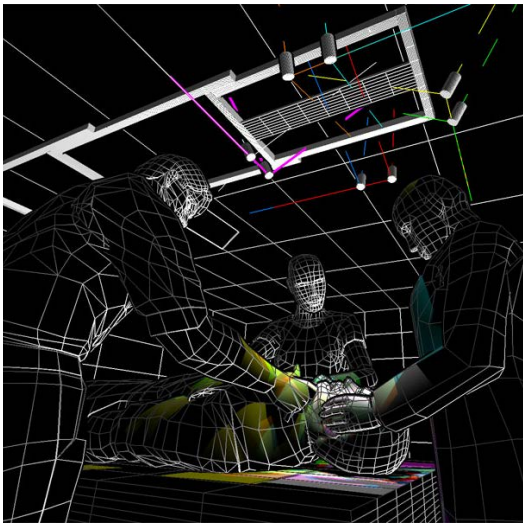
5. SIMULATING A 3D RECONSTRUCTION AND VISUALIZATION SETUP

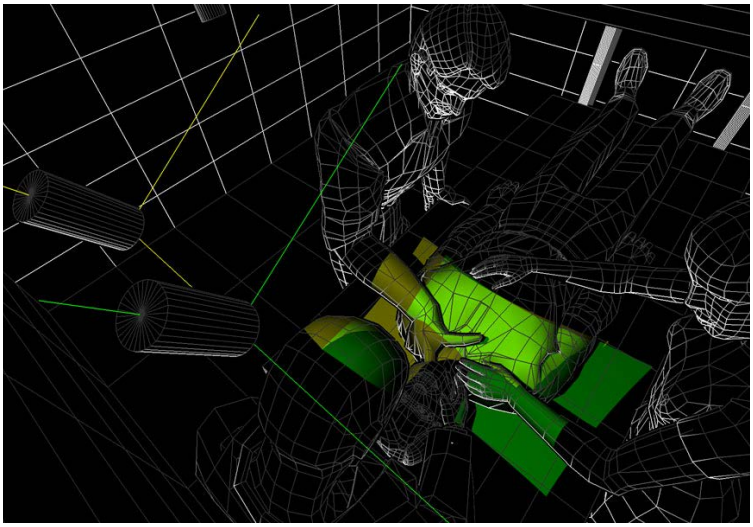
We put Pandora to work on the 3DMC project mentioned in the introduction. As already described, this project's long-term goal is live real-time 3D reconstruction of emergency medical procedures. The medical scenario we are currently working with (in collaboration with trauma surgery personnel at UNC hospitals) is *difficult airway management*, a critical respiratory condition which sometimes requires an emergency *cricothyrotomy* (incision into the trachea through the neck in order to enable the patient to continue breathing). Welch[16] provides additional details about the medical issues as well as about our group's preliminary research work on this topic.

We prepared a simple geometric model of a difficult airway patient together with three emergency medical personnel attending to him (Fig. 5, left). The main simulation task consisted in positioning video cameras on an existing mobile rig (inherited from a previous research project) such as to obtain adequate coverage of the spatial zone to be reconstructed. Additionally, we had to devise a remote viewing station for the live reconstructed data, and we had to establish geometric relationships between the reconstruction scene and the viewing station.

Our current on-line 3D reconstruction system[15] can simultaneously process up to eight live video streams. We simulated various camera arrangements, initially aiming for an octagonal setup (Fig. 5, center) similar to the one we used in the IEBooks project[14]. However, Pandora demonstrated that when medical personnel crowd around the patient, their heads and shoulders often occlude the relevant areas from the cameras (Fig. 5, right). After additional experimentation, we realized that we must image the treatment site through a single opening between two of the medical personnel. The most suitable (largest) such visual path is the open area above the patient's abdomen. After consulting with reconstruction experts, we eventually settled on a 4 (horizontal) by 2 (vertical) rectangular arrangement which attempts to exploit the available width of the opening, while still providing some vertical disparity (Fig. 6, left, and Fig. 7, left), and without encumbering the attending medics. The completed acquisition rig (Fig. 6, right) was built to match the measurements optimized in simulation.

Fig. 5. Pandora simulation for 3D Medical Consultation system. Left: pre-modeled geometry. Below left: Tentative octagonal camera arrangement. Below right: the medical personnel can easily block the cameras' view.





This arrangement is also advantageous for the remote viewing station. While our reconstruction system is capable of generating novel views of reconstructed geometry from arbitrary viewpoints, it (obviously) provides the best image quality for views originating close to or in between the acquisition cameras. It is therefore advantageous to position the acquisition grid at eye level and build a viewing setup that puts the virtual 3D image of the reconstructed structures at table height.

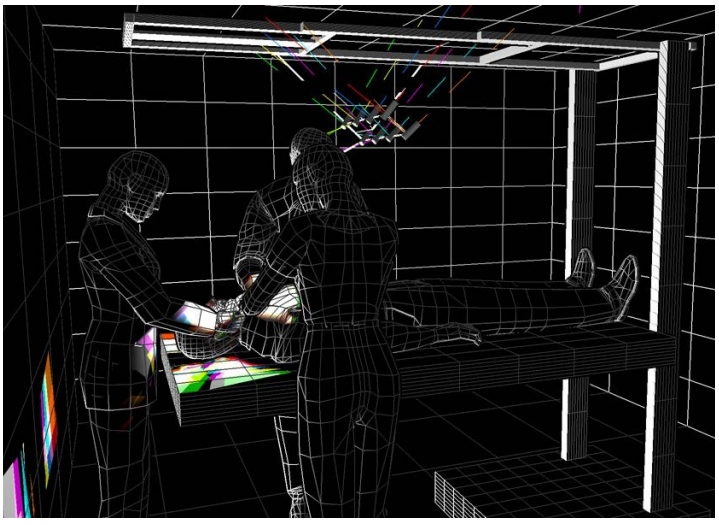
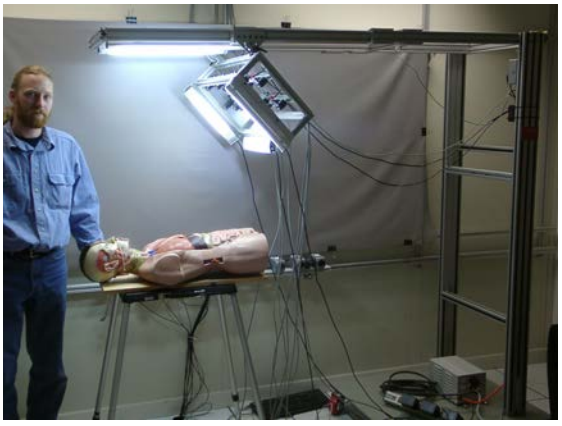


Fig. 6. Retained camera setup for 3D Medical Consultation prototype. Left: simulation. Below: acquisition rig.



Our initial remote viewing station was equipped with a single-viewer, head-tracked, non-stereoscopic, 20" flat panel display. Despite the (preliminary) absence of stereo visualization, we aimed to provide life-sized display of the reconstructed structures, for visual perception reasons. Furthermore, we wanted the remote consultant to be able to view the treatment site (patient neck and hands of the medical personnel) despite the limited size of the display. Once again it became apparent that a simple 3D simulation of the viewing station design would help with the exact geometric layout. Under the assumption that Pandora's geometric models of the acquisition-treatment site represent the virtual image of the reconstructed geometry, we added an interactively posable model of the flat-panel display into the scene. We extended Pandora to texture-map the display model with the view of the treatment site that would be visible from an equally freely movable viewpoint representing the remote consultant (Fig. 7, left). We positioned the display roughly perpendicular to the main axis from the 4-by-2 camera assembly towards the patient's throat. We then moved the display as close to the patient as possible (i.e., up to the point where we estimated that protruding virtual objects would be clipped by the sides of the display as the head-tracked observer standing in front of it moved laterally), while still preserving a large enough visual field on the display. To help with the development of the head tracking system, we also incorporated a freely movable model of the tracker unit (Origin Instruments DynaSight) together with an approximate wireframe display of its angular range. The completed viewing station (Fig. 7, right) was built to match the measurements optimized in simulation.

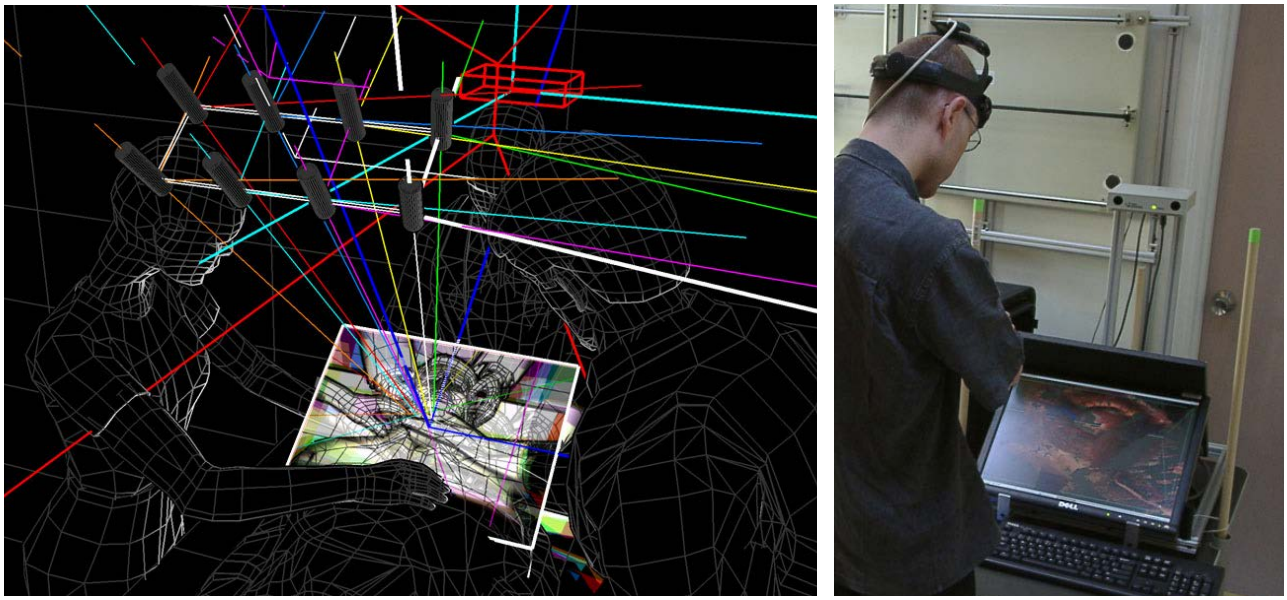


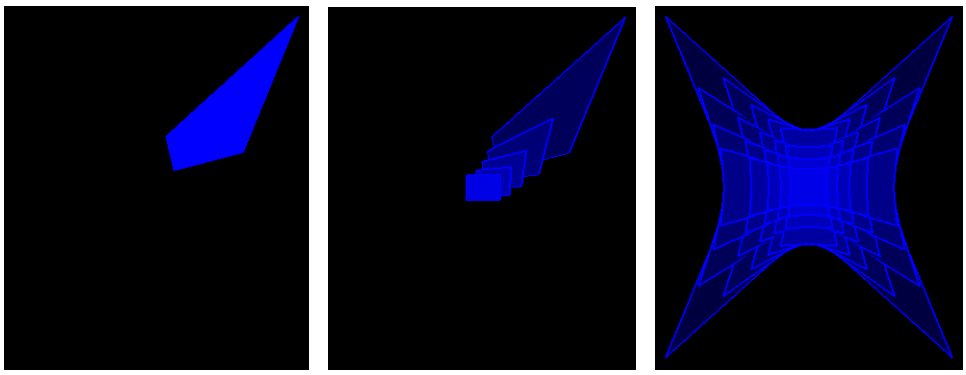
Fig. 7. Remote viewing station for 3D medical consultation. Left: simulated display; note 4-by-2 camera assembly and head tracker with acquisition range (red wireframes). Right: viewing station in use; note tracker (grey box w/ green LED above the display).

Welch[15] shows preliminary results obtained with the viewing setup described above and with a precursor of the acquisition rig in Fig. 6. We are currently designing and building a new remote viewing station equipped with a stereoscopic display. We will subsequently experiment with the complete pipeline (live multi-camera acquisition, on-line 3D reconstruction and real-time head-tracked stereo display) during a real medical procedure at the UNC trauma unit.

6. SIMULATING PAN-TILT-ZOOM CAMERA CONFIGURATIONS

When reconstructing dynamic scenes, there is no need to image the entire environment at every time step, because only parts of it change at any given time. Pan-tilt-zoom (PTZ) cameras have been used for surveillance in research efforts (e.g., work by Kanade[4]), where they mainly monitor the scene. Additionally, they can be used for more efficient 3D reconstructions that update only the changing parts of a scene. We extended our camera placement simulator to handle the placement of PTZ cameras and provide insights for their placement, both in the case of surveillance, and in the case of 3D reconstruction.

The designer of a camera system that contains PTZ cameras would benefit from knowing what regions can be covered by a PTZ camera placed in a given position, starting from a "neutral" orientation (0 degrees pan and 90 degrees tilt), and within a specified time interval. The configuration of a PTZ camera is characterized by its base position and orientation, and by values for its pan, tilt, and zoom settings. To show the camera coverage, we vary the pan, tilt, and zoom values, then generate the corresponding light pyramids, effectively forming *traces* that highlight the scene areas that the camera can cover. Since the projection of each pyramid requires a separate rendering pass, the system would quickly become too slow to be usable. However, under the simplifying assumption that PTZ cameras rotate around their center of projection, traces are formed of pyramids that share the same tip. The information contained in them can be assembled into a single pyramid of very wide angle. To account for the effect of each pyramid, we must highlight the corresponding part of the area covered by the wide-angle pyramid. We do this by rendering patches of areas proportional to the zoom values, warped to match the spherical coordinates specified by the pan and tilt angles. Fig. 9 (top left) shows an example of such a warped patch. We compute the appropriate patterns to be projected for each pyramid and accumulate them into a texture associated with the wide-angle pyramid.



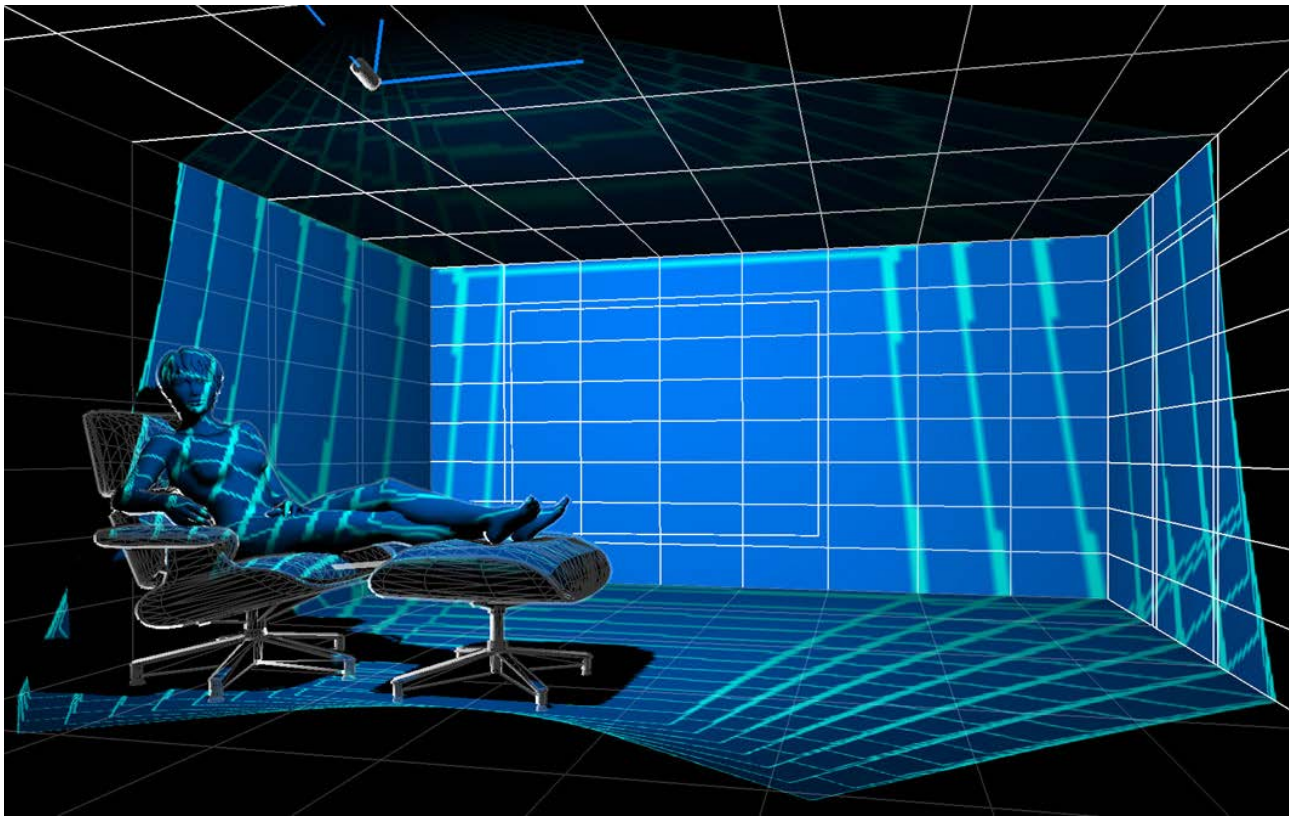


Fig. 9. Top row: Visualization modes of the simulator for PTZ cameras. Left to right: the "single", "path" and "all" modes (see text for details). Below: home security application example; a ceiling-mounted PTZ camera watches the window on the facing wall, but can still reach the doorways on the lateral walls within the user-specified interval.

The user can vary the base position and orientation of the PTZ camera using the joystick. The values of the pan, tilt and zoom parameters are updated using sliders in the GUI. The GUI also allows setting the motor speeds for each parameter, in degrees per second for the pan and tilt angles and in percent per second for the zoom. A separate light pyramid is shown for every one-second interval, so the speed settings effectively control how many intermediary light pyramids are shown on a trace. Additionally, the user can also specify the maximum length of the trace in seconds. The projected texture is recomputed when the user employs the GUI to change the camera parameters, but remains the same when the base position or orientation are changed using the joystick. To aid with the design of surveillance camera configurations, the GUI allows the user to quickly adjust the base orientation of a camera so that its neutral orientation is aligned with the current pan and tilt settings. This operation effectively re-centers the wide-angle pyramid to best cover the target area.

The simulator allows three visualization modes for PTZ cameras, shown in Fig. 9 as they appear reprojected on the wide-angle pyramid texture. The "single" mode shows the impact of a combination of pan-tilt-zoom settings. The "path" mode shows a number of intermediary positions between the neutral position and the position characterized by the current pan, tilt, and zoom settings. The "all" mode shows the total area that can be covered by a PTZ camera in a limited time interval. When showing multiple positions, we use darker colors to represent positions further in time from the present. Just as with regular cameras, the coverage of multiple PTZ cameras is illustrated using different colors for each camera.

We have set the angle of the wide-angle pyramid to 160 degrees in order to maintain a reasonable resolution of the details in the projected texture. This is a practical limit for the tilt angles of most cameras, as the mechanical base usually occupies the remaining 20 degrees. To illustrate the coverage of the entire domain for the pan angle (0-360 degrees), 3 wide-angle pyramids can be used per single PTZ camera, with an overlap of 20 degrees on each side.

CONCLUSIONS

We have shown how to simulate in real time multi-camera imaging configurations in complex geometric environments. Our interactive visibility simulator helps us assess in advance conditions such as visibility, overlap between cameras, absence of coverage and imaging resolution everywhere on the surfaces of a pre-modeled, approximate geometric dataset of the actual real-world environment the cameras are to be deployed in. We have applied our simulation technique to a task involving real-time 3D reconstruction of a medical procedure. It has proved useful in designing and building the multi-camera acquisition system as well as a remote viewing station for the reconstructed data. We have extended the simulator to support pan-tilt-zoom cameras, for which we can display coverage areas under consideration of the time interval required to reach them.

The visibility simulator is not an automation tool, but a planning aid requiring a skilled human system designer to interactively steer a simulated multi-camera configuration towards an improved solution. While we are a long way away from an automated optimizer, we expect our current tool to help designers develop intuition about the complex geometric relationships that govern multi-camera imaging systems.

Additional material accompanying this paper, including a video segment showing Pandora in action, is available from the project web site, which at the time of this printing is `http://www.cs.unc.edu/Research/nlm/pandora/`.

FUTURE WORK

Possible future improvements and advances include:

Better visualization for the simulation. This could be accomplished with the help of a stereoscopic display, possibly in combination with head tracking, or even through full viewer immersion into the simulated environment, by means of a motion-tracked head-mounted VR display. (It certainly makes sense for a simulator that simulates VR scenarios and displays to actually incorporate a VR display.)

Improved occlusion accuracy. This could be accomplished by enhancing Pandora with better hardware-accelerated algorithms for cast shadows, such as perspective shadow maps[12] or stencil buffer techniques[9].

Improved support for pan-tilt-zoom cameras. In practice, PTZ cameras do not rotate around their center of projection. In many cases, the center of projection and center of rotation are reasonably close, and the effects on the planning task are negligible. For the remaining cases, we plan to implement cameras with separate rotation and projection centers.

Animated geometric scenarios. As mentioned, the pre-modeled geometry used in the simulator can only represent an approximation of the real environment we want to capture. (If we had an accurate geometric model of that environment, why would we need to capture it again?). It is relatively easy to prepare such approximate models, which are nevertheless effective in pinpointing problematic camera setup issues. Increased effectiveness could be obtained with animated models. The trauma surgery scenario might benefit from insight gained when experimenting with animated medical personnel, since the motion of hands performing medical manipulations on the patient may interfere with the quality of the 3D reconstruction in areas of the patient's surface located below the moving hands.

Integration with camera calibration tools. Pandora already computes and simulates imagery from the point of view of its simulated cameras (Fig. 2, left). This imagery could be warped to simulate lens distortion. In addition, Pandora could be enhanced to simulate calibration targets such as checkerboard patterns. Thus equipped, Pandora could incorporate camera calibration algorithms and could be used to rapidly verify these algorithms' performance and suitability (with regard to geometric considerations, not photometric ones) for the camera configurations being evaluated, by graphically displaying the calibrated cameras' parameters (position, orientation, intrinsics) next to the known "ground truth," i.e. the simulated cameras.

Integration with reconstruction tools. Pandora could also serve as a testbed for 3D reconstruction algorithms. Synthetic imagery rendered from the simulated cameras' points of view, together with the calibrated camera parameters, could be used to attempt to actually reconstruct the simulated surfaces. The reconstruction algorithm's performance (again, only from a geometric point of view) would be assessed by displaying and inspecting the resulting reconstructed surfaces together with the known "ground truth," i.e. the pre-modeled geometry.

Integration of Pandora and Artemis. On specialized parallel machines, or on future high-performance hardware, Artemis and Pandora could be integrated into a single hybrid system, with Pandora acting as a user interface for Artemis' asymptotic stochastic evaluator. However, it is not unthinkable to assemble a somewhat more restricted, but still effective hybrid system based on contemporary hardware. It could use Pandora's hardware-accelerated evaluation of occlusion, visibility, and imaging overlap, and it would limit reconstruction uncertainty evaluation using Artemis' expensive algorithms to only those surface areas visible from more than one camera. Additional speedup could be achieved by implementing a user interface that further restricts stochastic evaluation to surface areas (or small volumes) manually selected by the user.

ACKNOWLEDGMENTS

B. Danette Allen (developer of Artemis) and Erica Stanley provided test cases for Pandora, stimulating the development of new capabilities. John Thomas and Jim Mahaney constructed the multi-camera rig visible in Fig. 6 (right). Mahaney also constructed the remote viewing station (Fig. 7, right). We would also like to acknowledge project collaborators Vincent Noel, Hua Yang, Herman Towles, Henry Fuchs, and Bruce Cairns, M.D.

We used *libwave* by Dave Pape to read and display the human models we set up with Curious Labs Poser 4 (now e frontier) and exported as alias|wavefront .obj format geometry files. The Charles Eames lounge chair with ottoman model was obtained from www.3dputty.co.uk. The simulator's graphical user interface was built using Xforms by T. C. Zhao and Mark Overmars. The simulator drives the double joystick controller through the Virtual Reality Peripheral Network (VRPN), developed by the NIH National Research Resource in Molecular Graphics and Microscopy at the University of North Carolina at Chapel Hill, supported by the NIH National Center for Research Resources and the NIH National Institute of Biomedical Imaging and Bioengineering.

Funding for this work was provided by the U.S. National Institutes of Health's National Library of Medicine (contract N01-LM-3-3514: "3D Telepresence for Medical Consultation: Extending Medical Expertise Throughout, Between and Beyond Hospitals").

REFERENCES

1. B. Danette Allen and Greg Welch. "A General Method for Comparing the Expected Performance of Tracking and Motion Capture Systems." *Proceedings of the 12th ACM Symposium on Virtual Reality Software and Technology* (VRST), Monterey, CA, November 2005, pp. 201-210.

2. Xing Chen. "Design of Many-Camera Tracking Systems for Scalability and Efficient Resource Allocation." PhD thesis, Stanford University, 2002.

3. Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

4. Takeo Kanade, Robert Collins, Alan Lipton, Peter Burt, and Lambert Wixson.. "Advances in Cooperative Multi-Sensor Video Surveillance." *Proceedings of DARPA Image Understanding Workshop*, volume 1, November 1998, pp. 3-24.

5. Jaron Lanier. "Virtually There: Three-dimensional tele-immersion may eventually bring the world to your desk." *Scientific American*, April 2001, pp. 66-75.

6. Benjamin Lok, Samir Naik, Mary Whitton and Frederick Brooks. "Incorporating Dynamic Real Objects into Virtual Environments," *Proceedings of ACM 2003 Symposium on Interactive 3D Graphics*, Monterey, CA, April 2003, pp. 31-41. Also: Images in *Communications of the ACM*, July 2003, 46.7, p.48.

7. Thomas Luhmann. "Rechnergestützte Planung und Berechnung von Aufnahmekonfigurationen für CCD-Bildsensoren." *Zeitschrift für Photogrammetrie und Fernerkundung* 3/1994, pp. 103-110.

8. Wojciech Matusik, Christopher Buehler, Ramesh Raskar, Steven Gortler, and Leonard McMillan. "Image-Based Visual Hulls." Proceedings of SIGGRAPH 2000, New Orleans, LA, July 2000. In *Computer Graphics* Proceedings, Annual Conference Series, 2000, ACM SIGGRAPH, pp. 369-374.

9. Morgan McGuire, John F. Hughes, Kevin T. Egan, Mark J. Kilgard and Cass Everitt. "Fast, Practical and Robust Shadows." http://developer.nvidia.com/, 2003.

10. Katri Elina Oksanen. "The Design and Simulation of Video Digitizing by Using Three-dimensional CAD-models." *International Archives of Photogrammetry and Remote Sensing*, Vol XXXI, Part B5, Vienna (ISSN 0256-1840), pp. 432-437.

11. Ramesh Raskar and Michael Cohen. "Image precision silhouette edges." *Proceedings of the 1999 symposium on Interactive 3D graphics*, Atlanta, GA, April 1999, pp. 135-140.

12. Marc Stamminger and George Drettakis. "Perspective Shadow Maps." Proceedings of SIGGRAPH 2002, San Antonio, TX, July 2002. In *ACM Transactions on Graphics*, 21 (3), pp. 557-562.

13. http://www.realviz.com/products/svz/index.php/

14. Greg Welch, Ruigang Yang, Sascha Becker, Adrian Ilie, Dan Russo, Jesse Funaro, Andrei State, Kok-Lim Low, Anselmo Lastra, Herman Towles, Bruce Cairns, M.D., Henry Fuchs, and Andy van Dam. "Immersive Electronic Books for Surgical Training." *IEEE Multimedia*, 12(3), July–September 2005, pp. 22-35.

15. Greg Welch, Diane Sonnenwald, Ketan Mayer-Patel, Ruigang Yang, Andrei State, Herman Towles, Bruce Cairns M.D., and Henry Fuchs. "Remote 3D medical consultation," in *Proceedings of BROADNETS: 2nd IEEE/CreateNet International Conference on Broadband Networks*, Boston, MA, October 2005, pp. 103–110, Omnipress.

16. Greg Welch, Ruigang Yang, Bruce Cairns, M.D., Herman Towles, Andrei State, Adrian Ilie, Sascha Becker, Dan Russo, Jesse Funaro, Diane Sonnenwald, Ketan Mayer-Patel,B. Danette Allen, Hua Yang, Eugene Freid, M.D., Andy van Dam, and Henry Fuchs. "3D Telepresence for Off-Line Surgical Training and On-Line Remote Consultation. Susumu Tachi, editor, *Proceedings of ICAT CREST Symposium on Telecommunication, Teleimmersion, and Telexistence*, The University of Tokyo, Japan, December 2004, pp. 113-152 and color plate (invited paper).

17. Lance Williams. "Casting Curved Shadows on Curved Surfaces." Proceedings of SIGGRAPH '78, Atlanta, GA, August 1978. In *Computer Graphics* 12, 3, ACM SIGGRAPH, pp. 39-42.

18. Ruigang Yang, Marc Pollefeys, and Greg Welch. "Dealing with Textureless Regions and Specular Highlights—A Progressive Space Carving Scheme Using a Novel Photo-consistency Measure." *Proceedings of the 9th IEEE International Conference on ComputerVision* (ICCV) 2003, Nice, France, October 2003, pp. 576-584.