Occlusion in Collaborative Augmented Environments

Anton Fuhrmann, Gerd Hesina, François Faure
and
Michael Gervautz

Department of Computer Graphics,
Vienna University of Technology

Abstract

Augmented environments superimpose computer enhancements on the real world. Such augmented environments are well suited for collaboration of multiple users. To improve the quality and consistency of the augmentation the occlusion of real objects by computer-generated objects and vice versa has to be implemented. We present methods how this can be done for a tracked user's body and other real objects and how irritating artifacts due to misalignments can be reduced. Our method is based on simulating the occlusion of virtual objects by a representation of the user modeled as kinematic chains of articulated solids. Registration and modeling errors of this model are being reduced by smoothing the border between virtual world and occluding real object. An implementation in our augmented environment and the resulting improvements are presented.

Introduction to the problem of occlusion and related work

One of the main advantages of using an augmented environment [AZ97, FE93] for collaboration as opposed to an immersive setup is the direct interaction of participants in reality. While the collaborators in an immersive setup always have to rely on more or less satisfying representations of each other, ranging from disembodied hands or heads to complete bodies visualized in plausible poses, users of an augmented scenario always are able to directly see each other and the interface devices they are using. This combination of reality and virtuality leads to the problem of correct occlusion between real and virtual objects, which of course does not exist in an immersive environment.

Even when using semi-transparent HMDs, where virtual objects only appear as transparent overlays over reality, wrong occlusion can hide gestures or facial expressions of participants. When applied in a video-based augmentation setup virtual objects can completely hide real objects if not occluded properly. Even discounting the importance of the social interaction, wrongly occluding virtual objects subject the users brain to conflicting depth clues: the parallax of the stereo-rendered objects gives a farther distance than the one inferred by the occlusion of real objects. This not only leads to misconceptions of spatial properties by the user, resulting in errors when trying to grab objects, but also increases eyestrain and the probability of motion sickness.

Influences of the display system

As already mentioned, the properties of the display system influence the severity of the occlusion problem, as outlined in the following table:

display system (example)	occlusion order	virtual object occluding real object	real object occluding virtual object
back-projection / screen based (CAVE[CR92], responsive workbench[KR94])		impossible	inherent
semi-transparent HMD (Studierstube [SZ98])		inherent	semi-visible / software solvable
video + immersive HMD (UNC [ST96b])		inherent	software solvable

In screen-based augmented environments - fishtank scenario - or projection based setups like the CAVE occlusion of virtual objects by real ones is simple and straightforward: real objects are always between the display surface and the eye and therefore always occlude virtual objects. This yields excellent results as long as no virtual object is placed between participants. The only exception for this would be the projection-based virtual office [RA98], where front projection is used. Since the virtual objects are projected on top of the real ones - normally on walls or desktops - real objects can be excluded from this projection by projecting nothing (displaying black) in the relevant portion of the display area. This could for example be used to exclude pictures or windows from being projected onto, but also - when using the proposed optical tracking mechanism - to exclude the users hands from being projected on. Nevertheless the occlusion of the display surface by real objects still manifest: users hands and arms for example may drop shadows on the surface, thereby occluding parts of objects virtually in front of the shadow.

When using HMDs, the display surface is always between the eye and real objects. Without further processing virtual objects always occlude real ones. The only difference - albeit only gradual - exists between see-through HMDs utilizing semi-transparent mirrors and immersive HMDs being fed video images by headmounted cameras: the first only overlays semi-transparent computer-generated images over reality while the second one may display completely opaque objects.

Since the only case where the occlusion problem is solvable we concentrate on setups where the displayed virtual objects overlay images of reality. This can of course also be a desktop-based system, where video images are displayed overlaid by graphics.

Influences of the tracking system

Tracking of users (heads), input devices and real objects to be augmented is a major task which influences strongly the quality of the augmentation. Immersive environments can tolerate discrepancies between reality and computer generated images which would be impossible to ignore in an augmented setup. Since in an immersive situation the user only relies on computer generated images for hand-eye coordination, errors between hand position in reality and projected hand/cursor position in the environment almost never lead to problems when interacting with the virtual environment.

In an augmented environment however, misalignment between tracked real objects and their representations in virtuality - which do not have to be visual representations: an input wand may only be represented by its "hotspot", the point in space where its function takes place - can cause severe problems for the user to operate in the environment. Additionally, lag between reality and the computer-generated environment is much more noticeable than in an immersive situation, since the position of virtual objects in respect to the real surroundings can be immediately compared. This results in "swimming" behavior of the virtual scene, which may also lead to motion sickness.

When addressing the problem of occlusion, another quality of the tracking system used matters: the ability to supply the simulation with additional information regarding the occluding objects. Ideally, we would like the tracking system not only supply us with the position and orientation of one or more reference points on the occluding object, but also to deliver complete geometric information which enables us to determine which parts of virtual objects to occlude.

This yields the following classification by tracking system:

Supplies only positional data	supplies additionally geometric information
Magnetic tracking Mechanical tracking Optical tracking using beacons [ST96b]	video tracking delivering depth map from stereo [WL95] video tracking delivering contour data [BE97] video range tracking [RA98] using invisible structured light laser range tracking

Tracking systems providing occlusion data

A very efficient and self-contained approach would be video-based augmentation using stereo headmounted cameras and a HMD. Stereo video data could be used for inside-out tracking of the users position and orientation and also to generate a dense depth map of the visible scene usable for occlusion. While this approach seems to be the most promising in respect to versatility, it has still to overcome some drawbacks. The computational complexity of the depth map reconstruction allows only coarse approximations of the scene depth to be computed in real-time. Wloka and Anderson [WL95] only produce coarse approximations of occlusion in acceptable time. This will probably be solved with increasing computing speed of future systems. Another drawback, which cannot be solved without additional tracking information is the line-of-sight problem: objects not visible in the video images cannot be tracked. While tracking of these objects is not needed for solving occlusion, tracking of the users hand outside the viewing frustum may be necessary. Especially when using two-handed interaction as we do when using the Personal Interaction Panel (PIP) [SZ97], the user must be able to use his proprioceptic sense alone to manipulate virtual input devices without relying on his hand-to-eye coordination. This manipulation outside the users field of view is one of the essential advantages of two-handed interaction and avoids display cluttering and occlusion by interaction elements.

Berger [BE97] presented a contour-based approach to occlusion which delivers outstanding results while using essentially only 2D image processing methods. This reduces the computational costs drastically, but also suffers from the line-of-sight problems mentioned above.

Approaches using special hardware, as for example laser range scanning devices, while providing excellent data are in most cases prohibitively expensive and suffer in many cases from line-of-sight problems similar to the ones cited above. This holds especially in a collaborative situation where users standing close together examine and object between them, as for example in [FU97].

Tracking systems providing only positional data

At the moment most virtual environments -immersive and augmented- use tracking systems providing only position and orientation. This includes commercial magnetic and mechanical tracking devices and advanced beacon-based optical tracking systems like the one developed at UNC [WA92] or the structured-light approach of [RA98].

Sophisticated optical tracking systems like this, or hybrid optical/magnetic solutions like the one used in [ST96b] deliver high precision tracking data but unfortunately no additional data usable for occlusion.

Registration problems

The problem of registering the users head position with the position of the virtual camera and the virtual objects with the real environment has already been covered extensively [AZ94, ST96a, TA96, WH95]. Here - like in the previously cited approach to occlusion by [BE97] - a 2D image-based approach presented by Kutulakos [KU98] seems to be an adequate solution for video-based systems.

The registration errors present us with especially annoying artifacts when occluding virtual objects. Slight errors produce a visible gap between virtual and real object *(figure 1)* or overlapping effects which occlude a slice of the real object that is supposed to appear in front of the virtual.

Since registration of the commonly used magnetical trackers is notoriously instable, these errors tend to appear in most augmented environments. We try to present a solution for reducing the visual impact of these artifacts.

Requirements for occlusion in "Studierstube"

The collaborative augmented environment we have developed allows multiple collaborating users to simultaneously study three-dimensional scientific visualizations in a "study room" - German: "STUDIERSTUBE". Each participant wears an individually headtracked see-through HMD providing a stereoscopic real-time display. The use of individual displays and head tracking allows providing stereoscopic, undistorted images for every user. The see-through property of the HMDs allows users to see each other and avoids the fear of bumping into obstacles.

Objects in Studierstube may appear - unlike in projection based display systems - both between and beside users. Interaction and collaboration within arm reach of the users is possible and supported by scientific visualization applications we have presented in [FU97] and [FU98]. Since the resulting scenarios lead often to situations where users heads, hands or bodies or the PIP had to occlude virtual objects, we had to find a method for efficient handling of these occlusions.

The properties of Studierstube and our chosen hardware and software setup presented us with the following requirements for solving the occlusion problem:

- **View independence supporting multiple users**
  Studierstube has to support multiple users. Occlusion methods therefore have to support a view independence of methods, ideally holding the computational cost proportional to the number of users.

- **Minimize number of trackers**
  Representation of users bodies has to be supported with a minimal number of additional trackers. Otherwise the tracking requirement of multiple users could quickly exceed our hardware capabilities and the users tolerance to setup overhead for use of the environment.

- **Minimize rendering overhead**
  Since occlusion is only annoying when it does not work, the tolerable overhead for correctly simulating it is relatively low. We have to keep rendering passes, additional geometry or image-processing to a minimum.

Since our environment Studierstube uses - like most virtual environments at the moment - commercial magnetic trackers we need another approach to the occlusion problem which does not rely on tracking of geometry or depth.

Occluding with phantoms

Since we are using a tracking system, which does not supply geometric information of occluding real objects, we have to acquire this information previously. This can easily be done by modeling or digitizing sufficiently precise representations of the objects offline and placing them inside the virtual scene.

We call these mockups of real objects "phantoms". By rendering them in an appropriate way occlusion of the virtual objects by real objects can be simulated *(figure 6)*. Rendering of phantoms can be performed differently, depending on the display system used:

Semi-transparent HMDs only need black areas where reality should be visible. To correctly occlude objects in this setup phantoms have therefore only to be rendered in black, without any special rendering order. This is fast and easy to implement.

When combining images for a video-based approach externally using a luminance- or chroma-keyed system the same method can be applied. For presentation purposes or when generating movies of our environment we usually apply digital compositing using one machine - an SGI Octane - for simultaneously digitizing video data and rendering virtual objects overlaying the video information.

The previous rendering approach would only produce a black avatar in front of the video image. To correctly render occluding geometry we had to restructure the scenegraph to guarantee all phantom would be rendered before any visible geometry. This can be done in a pre-processing step and does not influence rendering times. The following OpenGL sequence is executed for each frame:

reset z-buffer
render video background
enable z-buffer testing & writing, disable RGB writing
render phantoms
enable RGB writing
render visible part of scene

This results in "invisible" phantoms, which only are registered in the z-buffer of the rendering hardware. Normal geometry is only rendered where it lies nearer to the viewpoint as a phantom.

Occlusion of static objects

The simplest case of occlusion of real by virtual objects is when the real objects are previously known and static over the duration of the simulation. Examples for this would be furniture and fixtures of the room used as workspace.

Examples for occlusion of static objects can be found in [BR96]. We have applied this method to simulate occlusion of virtual objects by laboratory furniture in Studierstube.

Occlusion of tracked rigids

The occlusion of virtual scenery by non-stationary rigid real objects can be handled in much the same way as the above case of static objects. The position and orientation of the phantom - since they are no longer constant over the duration of the simulation - have to be coupled to

the real object, in our setup by the mounting of an additional magnetic tracker sensor on each occluding object.

We have used this approach to model occlusion of virtual objects by tracked real objects like the PIP in Studierstube *(figure 6)*.

Occlusion for tracked articulated objects

A more general case than tracking simple rigid objects for occlusion purposes is the use of tracked articulated objects for occlusion. The primary application of this in our environment was the occlusion generated by participant's bodies moving in front of virtual objects.

To achieve this under the requirements stated above we modeled a coarse articulated representation of a user. This approximation is used as a "phantom avatar", which is supposed to mimic all poses and gestures of the tracked user to allow rendering of correct occlusion. It consists of rigid segments corresponding to body parts, which have to be animated in real-time according to incoming tracker data. To achieve this we have to mount additional trackers on the user. *(figure 5)* shows how this was done. Since the head position is already tracked via the HMD, only three additional tracker sensors were used: two on the lower arms and one on the back of the user, a scheme comparable to the one used by Badler [BA93]. Similar results have been obtained by Waldrop [WA95] with the use of three sensors per arm for implementing avatars in a distributed virtual environment.

Based on a reduced number of sensors our simulation has to implement a number of additional constraints regarding joint stiffness and degrees of freedom to correctly simulate the posture of the user. Since our system is based on rigids we also are not able to simulate deformations of the body, but are planning to implement this in a later version based on a simplified, segmented spine.

Compensate registration errors by "blurring" the phantom

As already stated before, slight registration errors between phantom and real object result in annoying artifacts when occlusion occurs. This can be caused by errors between tracker data and real position of the sensor, due to time lag or misregistration of real and virtual environment or by differences between the shape of the user and phantom geometry. The first two causes can be addressed in one of the ways cited above, whereas the last case defines a new problem.

While theoretically possible, a pixel-perfect representation of for example the users forearm may be possible to model, such precision is in most cases unnecessary. Even discounting remaining errors from the tracker misregistration deformation of body and even changed clothes can invalidate the precision in modeling the shape of a body part in a fixed posture. Furthermore a precise and therefore finely tessellated model of the avatar would consume too much of our polygon budget.

We are therefore proposing to represent this margin of error in our phantom avatar. We do not render a hard edge between occluded object and occluding phantom but a soft transition covering the error margin *(figure 3)*. This transition shall enable the user to perceive details of both virtual object and occluding real object, while not significantly reducing the visual occlusion clue.

A further application of this transitional "blurring" effect are body parts where much additional tracking effort would have to be spent just to implement precise occlusion: the users hands. Without using the almost always cumbersome and inaccurate gloves to track exact hand and finger positions, not enough data is available to correctly occlude this part of the users body. When using some kind of "probability blurring" in this area, which renders the avatar the more occluding, the higher the probability of a part of the hand appearing in it is, we are able to provide satisfactory occlusion (figure 2). Since finger and hand gestures are

desirable extensions of collaboration in our environment we want each user be able to see them.

Implementation

The Studierstube-setup

As mentioned before, each participant in our collaborative augmented environment STUDIERSTUBE wears an individually headtracked see-through HMD providing a stereoscopic real-time display.

The HMDs we use - Virtual I-O i-glasses - are very lightweight and unobtrusive, but only of limited resolution and small (30°) field of view. As input devices we use a 6DOF pen with multiple buttons and the Personal Interaction Panel (PIP) [SZ97]. The PIP allows both intuitive three-dimensional manipulations such as placement of objects and the input of numerical data or commands via traditional 2D sliders and buttons *(figure 6)* - historically one of the weak points of VEs. The pen, the PIP and the HMDs are tracked by an ASCENSION flock of birds magnetic tracker.

Integration of the necessary extensions for occlusion into Studierstube proved to be relatively straightforward. The whole system is based on the OpenInventor [ST92] Rendering/Simulation library, which reduces the addition of new functionality to the Studierstube environment to the simple process of adding a new dynamically loadable module. The necessary tracker data is delivered by the Studierstube tracker interface, which delivers positions and orientations in OpenInventor fields, a high-level interface element, which allows easy connections to the separately developed simulation library [FA98].

Kinematic simulation of avatar posture

The phantom avatar used for occlusion is modeled as articulated solids representing a human body. A geometry used for occlusion is attached to each solid. Some of the solids are directly connected to the trackers attached to the user, as illustrated in figure 5.

Dynamics is used to animate the avatar. This allows us to intuitively rigidify the degrees of freedom of the joints using damped springs. An upward force applied to the body maintains a standing position. The resting positions of the springs define the natural, comfortable posture of the avatar.

Our simulator, detailed in [FA98], solves the inverse kinematics problem while taking into account masses and forces. Forces are applied at the beginning of each time step. The motion is then integrated over time, leading to new positions. In addition, the position of the solids bound to the trackers are set accordingly to the input data, regardless of forces or velocities. Possible violations of geometric constraints induced by forces, time integration and tracker input are then cancelled by computing appropriate displacements of the other solids.

The displacements are computed by solving the dynamics equation

$$(JM^{-1}J^{T})\lambda = b \qquad (1)$$

where matrix M is a block-diagonal matrix which represents the mass of the system and J a sparse matrix which codes for the geometric constraints. Vector b represents the geometric errors to cancel. Vector λ gathers Lagrange multipliers of the constraints. They act much like integrated forces. The corresponding motion corrections are are thus $M^{-1}J^{T}\lambda$. Velocity updates are deduced from position corrections by applying Stoermer's rule [PR92]. This integration scheme considers displacements instead of velocities. This avoids us to differentiate the possibly noisy tracker data.

An iterative solution of equation (1) is performed. A minimization of the norm of the error is performed over the search space defined by the components of vector λ. This approach has two useful features for our application. First, in presence of overconstrained solids, such as the upper arms of the character in figure 5, a compromise between the constraints is found. Second, the iterative minimization allows the user to trade-off accuracy for computation time, making the approach suitable for the real-time animation of complex scenes.

Implementing "blurred" phantoms

To implement the "blurring" we want to exhibit the avatars, we have essentially two approaches of realizing this effect:

Image-based blurring

The first method that comes to mind when an operation like "blurring" is requested is of course an image-based approach. Rendering an avatar blurred by some kind of convolution operation seems to be the simplest way to implement the desired effect.

But a normal convolution using only software for a video-resolution image may not be completed in real-time (e.g. for frame rates >10Hz) without the use of special hardware. Convolution techniques utilizing OpenGL hardware as presented in [MC98] may be fast enough, but depend in our case on the ability of moving data fast between image buffer and texture buffer, which is not present in all OpenGL [WO97] implementations. Using the OpenGL accumulation buffer would provide an elegant solution for this problem, but unfortunately would require N² rendering passes of the whole scene for an N×N convolution matrix.

Furthermore certain properties of the image-space approach may not result in the desired appearance of the occlusion: we want the margin of error to be specified in absolute, real-world measurements, to allow for a fixed error produced by e.g. different clothing or hairstyles. An image-based approach would deliver the same amount of blurring for near and far parts of the avatar. The above mentioned "probabilistic blurring" would also not be easily implementable using a single convolution over the avatar.

Object-based blurring

Another approach to the implementation of smooth transitions for occlusion is to implement a "blurred geometry" of the avatar. This consists not of a single surface representation of the geometry representing a body part, but a layered approach of successively bigger and more transparent shells around the geometry *(figure 2)*. These shells can be modeled as the same geometry rescaled around the centroid of the object, since most of the avatars geometry is convex.

Of course this approach does not yield a "smooth" transition, but one consisting of a discrete number of steps in occlusion, but it should be sufficient to implement a reasonable small number of steps - say three to ten - to obtain satisfying results in most cases. Since this geometry undergoes the same perspective transformations as everything else, the error margin can be specified in real-world measurements as opposed to the image-based approach above.

A further advantage of this method is the ability to implement "Levels of Detail" for phantoms: with increasing distance we can "switch off" more and more of the intermediate shells, to reduce geometric complexity and therefore polygon count for objects farther away. This allows us to vary the number shells and of transition steps according to distance. In distant phantoms only one or two transitions, covering maybe an equivalent amount of pixels of distance between two shells in image space are rendered. Phantoms nearer the observer are represented by more shells, resulting in approximately the same number of pixel per transition step. This behavior is readily implemented using the standard OpenInventor LOD-node,

which references more or less of the scaled versions of the same geometry representing the avatar.

Evaluation and Results

Registration of the avatar geometry to the user proved to be a relatively time consuming process. Since misregistrations of the tracking sensors' position relatively to the geometry resulted not only in visual artifacts (figure 1) but also in artifacts of the simulation, registration had to be done very carefully. Once registered, the main problem was an additional time-lag introduced by the kinematic simulation, which could only partially be corrected in this version. The visual results in a typical scientific visualization situation as in (figure 3) were satisfying and gave users a much better overall impression of the situation and the spatial relationship between real and virtual scene.

The object-based implementation of "blurred" occlusion was easy to implement and gave in the cases where hand-gestures made rigid modeling difficult a satisfactory compromise between correct occlusion and tracking effort (figure 2 and 3). It produced noticeable artifacts along the intersections between rigid limbs, which can be corrected by using stencil masks when rendering the different shells, a correction which will be implemented in the next version of our occlusion system.

The main difficulty for the animation of the avatar is to have it accurately overlap the image of the user (figure 7). As mentioned before, the geometry has to be carefully adjusted to the body of the user (figure 8). Additionally, the positions of the trackers and of the joints with respect to the geometry have to be precisely modeled. This is done interactively before performing the animation. Joint positions are difficult to adjust because they are inside the body of the user, thus invisible. We adjust them iteratively using different postures.

There are several sources of error in the final result (figure 9). The geometric models do not fit perfectly to the body of the user. The human body is a much more complex structure than our avatar. Limbs are not rigid, especially when covered with cloth, and the joints of the human body differ from our simple joints. If the calibration of the virtual camera does not fit perfectly to the real camera used for video input, the avatar may overlap the user in some postures but not in others. Future work include a more realistic body structure, especially for shoulders, back and neck, and semi-automatic calibration of the virtual camera.

Future Work

We are planning to introduce a calibration scheme for rescaling the phantom avatar to fit different users. The basic idea for this is to let the user strike a small number of predefined poses, which should enable the system to infer body measurements.

We plan eventually to get rid of the sensor on the users back by additional constraints in the head/back joint and to introduce certain assumption of the body posture (vertical) for our preferred work situations, standing and seated.

A minimum sensor situation, using only PIP and pen and the information on the users handedness (right- or left-handed) to infer avatar posture is being looked into.

For further information contact via email
fuhrmann|hesina|faure|gervautz@cg.tuwien.ac.at or access the latest results
on http://www.cg.tuwien.ac.at/research/vr/occlusion/

References

[AZ97]   Ronald Azuma. A Survey of Augmented Reality. Presence, 6(4):355-385, 1997

[AZ94]   Ronald Azuma and Gary Bishop. Improving Static and Dynamic Registration in anOptical See-Through HMD. Proceedings of SIGGRAPH'94, pages 197-204, 1994.

[BA92]   M. Bajura, H. Fuchs, and R. Ohbuchi. Merging Virtual Objects with the Real World: Seeing Ultrasound Imaginery within the Patient. Proceedings of SIGGRAPH'92, (2):203-210, 1992.

[BA93]   Norman I. Badler, Michael J. Hollick and John Granieri. Real-time control of a virtual human using minimal sensors. Presence 2(1), 1993, pp. 82-86.

[BE97]   M.O. Berger. Resolving Occlusion In Augmented Reality: A Contour Based Approach Without 3D Reconstruction. Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, 1997, Poster Session 1.

[BR96]   David E. Breen, Ross T. Whitaker, Eric Rose and Mihran Tuceryan. Interactive Occlusion and Automatic Object Placement for Augmented Reality. Computer Graphics Forum (Proceedings of EUROGRAPHICS'96), 15(3):C11-C22, 1996.

[CR92]   C. Cruz-Neira et al. The CAVE: Audio Visual Expirience Automatic Virtual Environment. Communacations of ACM, 35(6):65,1992.

[FA98]   Francois Faure. Interactive solid animation using linearized displacement constraints. Proceedings of the Eurographics Workshop on Animation and Simulation, 1998.

[FE93]   S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-Based Augmented Reality. Communications of the ACM, 36(7):53-62, 1993.

[FU97]   A. Fuhrmann, H. Löffelmann, and D. Schmalstieg. Collaborative Augmented Reality: Exploring Dynamical Systems. In IEEE Visualization '97 Proceedings, pages 459-462. IEEE Computer Society, October 1997.

[FU98]   Anton Fuhrmann, Helwig Löffelmann, Dieter Schmalstieg, and Michael Gervautz. Collaborative Visualization in Augmented Reality. IEEE Computer Graphics and Applications, 18(4):54-59, July/August 1998.

[GE92]   D. Geiger, B. Ladendorf and A. Yuille. Occlusions and Binocular Stereo. Proceedings of Computer Vision (ECCV '92), pp. 425-436, 1992.

[KR94]   W. Krueger and B. Froehlich Visualization Blackboard: The Responsive Workbench (virtual work environment) IEEE Computer Graphics and Applications, 14(3), pp. 12-15, May 1994.

[KU98]   Kiriakos N. Kutulakos and James R. Vallino. Calibration-Free Augmented Reality. IEEE Transactions on Visualization and Computer Graphics, 4(1), pp. 1-20, January 1998.

[MC98]   Tom McReynolds, David Blythe, Brad Grantham, and Scott Nelson. Advanced Graphics Programming Techniques Using OpenGL. SIGGRAPH `98 Course notes

[PR92]   Press, Teukolski, Vetterling, and Flannery. "Numerical Recipes in C", Cambridge University Press, 1992.

[RA98]   Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin and Henry Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. SIGGRAPH 98 Conference Proceedings, Annual Conference Series, pp. 179-188, Addison Wesley, July 1998.

[ST92]   P. Strauss and R. Carey. An Object Oriented 3D Graphics Toolkit. Proceedings of SIGGRAPH'92, (2):341-347, 1992.

[ST96a]   Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior Augmented-Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. Proceedings of SIGGRAPH 96, Annual Conference Series 1996, ACM SIGGRAPH, pp. 429-438.

[ST96b]   Andrei State, Mark A. Livingston, Gentaro Hirota, William F. Garrett, Mary C. Whitton, Henry Fuchs, and Etta D. Pisano (MD). Technologies for Augmented-Reality Systems: realizing Ultrasound-Guided Needle Biopsies. Proceedings of SIGGRAPH 96, Annual Conference Series 1996, ACM SIGGRAPH, pp. 439-446.

[SZ97]   Zs. Szalavári, M. Gervautz: The Personal Interaction Panel - A Two-handed Interface for Augmented Reality. Proc. EUROGRAPHICS 97, Budapest, Hungary, 335-346, 1997.

[SZ98]   Zsolt Szalavári, Dieter Schmalstieg, Anton Fuhrmann, and Michael Gervautz. Studierstube - An Environment for Collaboration in Augmented Reality. Virtual Reality: Research, Development & Applications, (3):37-48, 1998.

[TA96]   T. Oishi and S. Tachi. Methods to Calibrate Projection Transformation Parameters for See-Through Head-Mounted Displays. Presence, 5(1):122-135, 1995.

[WA92]   Mark Ward, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs. A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems. Proceedings of 1992 Symposium on Interactive 3D Graphics (Cambridge, Mass., March 29 - April 1 1992), 43-52

[WA95]   M. Waldrop, Marianne S., Pratt, Shirley M., Pratt, David R., McGhee, Robert B., Falby, John S. and Zyda, Michael J. Real-time Upper Body Articulation of Humans in a Networked Interactive Virtual Environment. Proceedings of the First ACM Workshop on Simulation and Interaction in Virtual Environments, University of Iowa, 13 - 15 July 1995, pp. 210-214.

[WH95]   R. Whitaker, C. Crampton, D. Breen, M. Tuceryan, and E. Rose. Object Calibration for Augmented Reality. Proceedings of EUROGRAPHICS'95, pp. 15-27, 1995.

[WL95]   Matthias M. Wloka and Brian G. Anderson. Resolving Occlusion in Augmented Reality. Proceedings of Symposium on Interactive 3D Graphics, ACM SIGGRAPH, 1995, pp. 5-12.

[WO97]   Mason Woo (Contributor), Jackie Neider, Tom Davis, Opengl Architecture Review board, "Opengl Programming Guide : The Official Guide to Learning Opengl, Version 1.1", 2nd edition (January 1997), Addison-Wesley Pub Co; ISBN:201461382
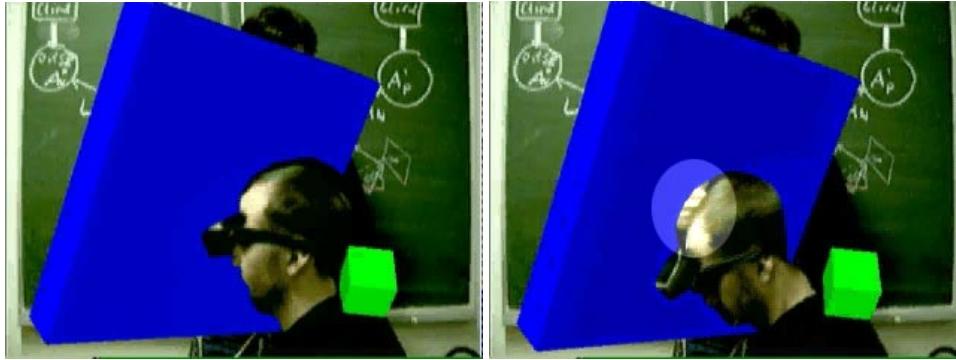
fig 1: Occlusion with head phantom, misregistration (in circle) due to time lag.
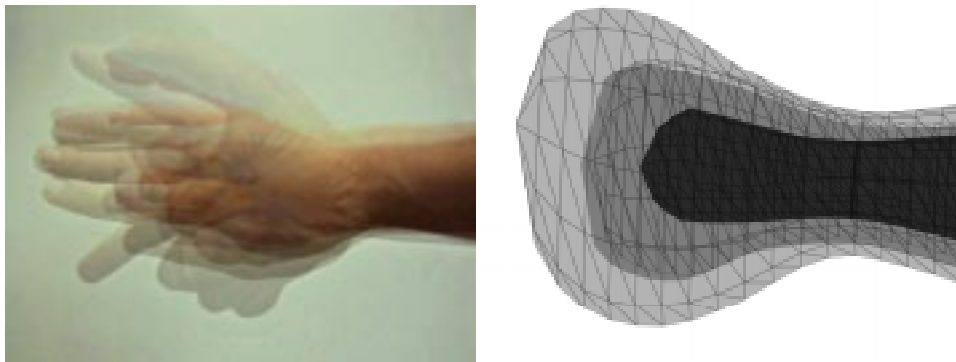

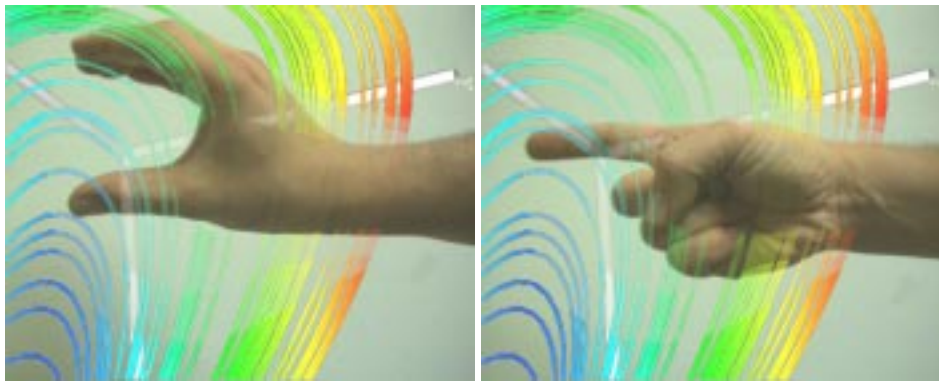fig 2: "Probabilistic" function of hand and corresponding phantom.

fig 3: Use of the same "probabilistic" phantom with different gestures.

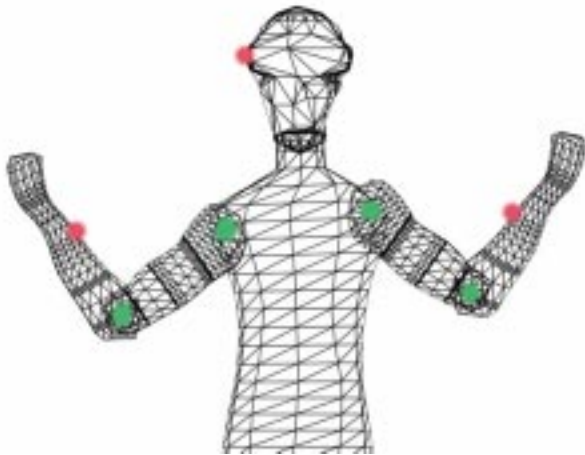fig 4: User with overlaid phantom avatar geometry.

fig 5: Phantom avatar with position of tracker sensors (red points) and joints (green points).
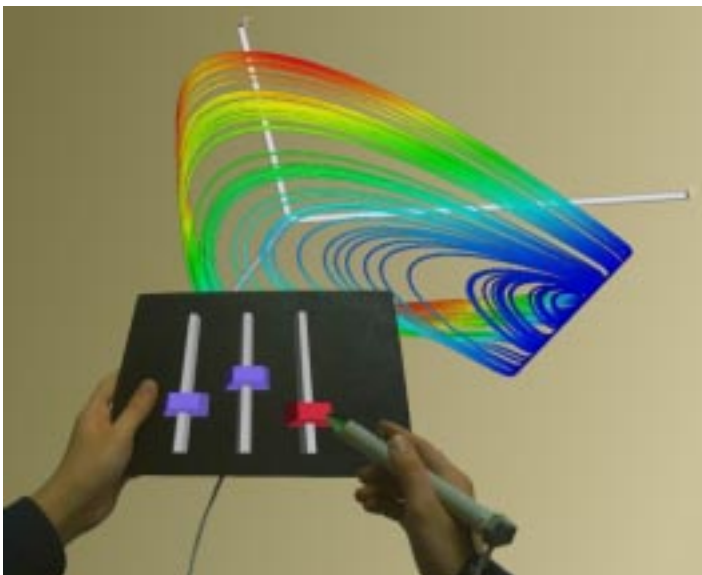

fig 6: Personal Interaction Panel (PIP) occluding scientific visualization.

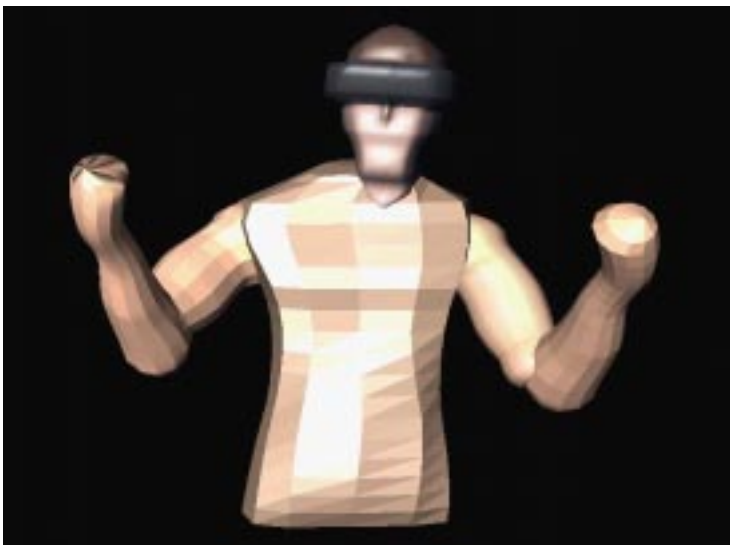fig7: User with 4 sensors and HMD.

fig 8: User represented as avatar.

fig 9: Occlusion of a background polygon.