

**Final Technical Report
Walkthrough Project**

June, 1992

to

**Computer and Information Science and Engineering
National Science Foundation**

TR92-026

**Frederick P. Brooks, Jr.
Principal Investigator**

**John Airey
John Alspaugh
Andrew Bell
Randolph Brown
Curtis Hill
Uwe Nimscheck
Penny Rheingans
John Rohlf
Dana Smith
Douglass Turner
Amitabh Varshney
Yulan Wang
Hans Weber
Xialin Yuan**

**Department of Computer Science
University of North Carolina
Chapel Hill, N.C. 27599-3175**

Six Generations of Building Walkthroughs –

Final Technical Report to the National Science Foundation

1. Executive Summary

From 1986 until the present a small team of faculty and students at UNC-CH have built a succession of interactive computer graphics systems to enable a viewer to experience an architectural model by simulating a walk through the model.

This work has been marvelously aided by, and has also driven, the work of Henry Fuchs, John Poulton, and the entire Pixel-Planes team, in the design and system programming of a series of experimental state-of-the-art graphic engines, Pixel-Planes 4, Pixel-Planes 5, and now PixelFlow. They have been very responsive to our needs.

Whereas Walkthrough-Ikonas was very crude, hindered chiefly by an update rate of only one picture per three seconds, and flat-shaded polygonal walls, the current Walkthrough 5.2 has attained a level of naturalness we believe to be unequalled anywhere in the world. We achieve an update rate of 25 stereo frames per second, or 50 ups total, on models of 30,000 polygons. The models are illuminated via radiosity calculations, with individual patches Gouraud-shaded. Walls, floors, ceilings, and furnishings may be textured, which greatly adds to realism. Sounds proper to each location are played as one walks about our house model. The user interface has reached new levels of naturalness. The user wears a stereo head-mounted display. One moves about within a 12'x10' space by ordinary walking, bending, stooping, etc.; and one can translate that space to any location in the house or yard by button-pushing. One can reach out a hand and open doors, pick up and carry certain objects, and shove furniture around. Figure 1 summarizes the capabilities added to our Walkthrough systems through this evolutionary period.

We have learned many lessons over the six years. The most important is that in any Virtual Reality system, as the Walkthroughs are, many of the technical tasks are much more difficult when one uses models of meaningful complexity, consisting of tens of thousands of primitive elements. Developing and maintaining a model of this size is an effort comparable to developing and maintaining a program of such a size, and the software engineering techniques necessary for big programs are indeed appropriate for, and necessary for, model engineering.

Moreover, achieving real-time performance with reasonably realistic images of models of the complexity of real buildings requires both very fast hardware and new rendering algorithms. It is, for example, absolutely necessary that one not render polygons that cannot possibly be seen from the current vantage point, and that requires dynamic partitioning algorithms not required for smaller models.

Naturalness for the user requires not only visual realism and motion realism, but a handy and natural interface, and as builders all kinds of computer systems have come to recognize, these require a great deal of work, much iteration and testing, and a great deal of care.

2. Objectives

The principal purpose of this research has not been the end product itself. We have at no point let our research be directed by either distributability considerations, such as whether our graphics engine would ever be available commercially, nor marketability considerations, such as limiting cost to a level one could afford to sell. Instead, it has been our purpose to build a realistic simulation that would

- drive all dynamic graphics technologies to the utmost, and
- be within the everyday experience of our users, so that they and we can at all times identify the system shortcomings, the defects that most impair the illusion of real presence.
- teach us about the behavior of people in simulations.

2.1 Real time

Our overriding objective has been to achieve motion realism, so that both the *refresh rate* and the *lag* between a user motion and the consequent change in the visual image presented are good enough not to hinder the illusion of real presence. We find refresh rates of 15 ups to be satisfactory. We do not yet have a satisfactory lag, and we do not know from either literature or experiment how short it will have to be. It turns out to be *much* more demanding to require a virtual object to stay still in virtual space as one moves the eyepoint, than to achieve a lag that is completely satisfactory in through-the-window graphics.

2.2 Realistic images

As soon as we achieved 15 ups, we began to concentrate on making images that are more visually realistic. We do not expect ever to be able to fool anyone as to whether what they are seeing is natural or artificial; we suspect that photorealism may be beyond the reach of synthetic processes. Nevertheless, it is a wonderful objective, and striving for it teaches us many things. Fortunately, a major part of the graphics field goes on before us in this quest, developing ever better techniques for visual fidelity, with computation-per-frame under no constraint. Our task is to judge the most powerful of these techniques and to adapt them to approximations computable within 1/15th second.

2.3 Real buildings

We originally lashed together a prototype Walkthrough system, and a model of the main lobby of the planned Sitterson Hall, in order to make a design decision about the width of the upper lobby outside the main desk. When we saw that exercise to yield many interesting research problems in real-time computer graphics, we formalized the project, and undertook it in dead earnest. From our experience with the Sitterson lobby, we determined to face the size and complexity challenges of real buildings.

Most of our modeling effort has been focused on three real buildings, each of which was modeled in its entirety, and before it was constructed: Sitterson Hall, the home of the Computer Science Department, the Fellowship Hall and Sunday School Wing of the Orange United Methodist Church, and a projected remodeling of the Brooks house. We also picked up Cornell's model of their new Theory Building and used Walkthrough 4 to enable its exploration.

For test purposes we have also modeled two offices, Sitterson 020 and 365 in furnished detail within the Sitterson models, and we built a quite detailed model of a four-room imaginary department in order to show our system more conveniently at SIGGRAPH 91. It turned out that this model, although visually very appealing, kept raising distracting questions as to why anyone would design the space that way. In fact, no one would; it was an artifice, and in the event we did not use it to demonstrate our system. This unhappy experience with an imaginary building, and our desire to be able to continually compare our simulated system with the real world, confirmed our resolution to stay with models of real buildings.

2.4 Natural navigation

Ideally, one would like to be able to:

- walk around a building-sized space, or an outdoors building site,
- look in any direction,
- know where one is in the virtual space
- collide with and be constrained by walls,
- climb stairs,
- open and close doors,
- turn lights and water on and off, and
- move furnishings and utensils about while acting out life routines.

Clearly some of these effects are very hard to simulate.

Physical motion powerfully aids the illusion of presence, and actual walking enables one to feel kinesthetically how large spaces are. Indeed, one would like to be able to walk about a proposed building site, seeing both the proposed building and the trees, shrubs, grass, etc. that will be there when it is finished, against a backdrop of the undisturbed natural features.

Although we still are far from this ideal, we have kept this natural navigation objective before us and have made substantial strides towards its achievement.

3. Six generations

Figures 2-5 show the six readily discernable generations, each built on the one before, each qualitatively different in the effectiveness of the illusion. We shall just highlight the illusion-boosters; the figures show contemporaneous advances on other dimensions. The details of each new system idea are discussed in Section 4.

3.1 Walkthrough I – Ikonas

Walkthrough I ("I" for *Ikonas*) in Figure 2 was our first real system, built on our feasibility-demonstration prototype, but honed for a year to get maximum speed and a really convenient user interface. Doug Turner built a complete single-wall map of Sitterson Hall, equipped one room with furniture, and experimented with lighting effects. Even though images took three seconds to generate, the dynamic tracking of the eyeball cursor on the floor plan made it easy to move to a position and view direction, then wait for the image to be

painted. Or, one could move smoothly down a hallway, with a new image appearing every three seconds.

Some of the interface strengths have not been re-realized on the Pixel-Planes system, but we need to pick them up again. The most important is the floorplan overview. Another, which must be done in a more sophisticated manner, is the sun direction control.

3.2 Walkthrough 4.1 - Pixel-Planes 4

Just re-implementing the system for Pixel-Planes 4 when it appeared was a years work, but it gave a 9-fold speed advantage, giving us 3 ups! Immediately we observed that people seemed to be able, with this degree of dynamism, to navigate about Sitterson by looking at the scene, and did not have to have a map, so we did not build the auxiliary floorplan guide. Figure 3 shows the three generations of systems built on PxPI4.

3.3 Walkthrough 4.2 - Model partitioned

The qualitative difference here was partitioning the Sitterson model into cells, manually, so that not all the polygons had to be rendered on each frame. This gave a factor of 5 in speed, bringing it to 15 ups. Real-time motion at last!

Other illusion-boosters in this generation include anti-aliasing, which quite noticeably enhanced image quality, the big wall screen (4'x6'), the treadmill as a navigation device, and two new big models – a double-walled model of Sitterson, and a model of the Orange Church Fellowship Hall Addition.

3.4 Walkthrough 4.3 - Radiosity

This generation showed a striking difference in visual realism, through the use of adaptive radiosity. The radiosity calculations themselves took hours for a big model, and even the run-time superposition calculation took about a second, so we used sweetening. When one stopped walking, the shadows fled away to the corners, and the lighting looked great.

Airey completed our basic software package with his automatic partitioning algorithm. We learned that displaying anything less than all visible cells is disconcerting. As you turn your head about the room, suddenly you see an open portal with no image in it. This is bothersome, even when the black space is filled with a lovely Carolina blue.

Also notable in this generation was our use of the head-mounted display, even though we didn't have the power to generate stereo images. Used in conjunction with the treadmill, it enabled the user to walk about and look about independently, quite a natural interface.

3.5 Walkthrough 5.1 -Pixel-Planes 5 and Stereoscopy

Pixel-Planes 5 provided an initial speedup of 8 to 10 when the first rack came on line, enough to enable us to get stereoscopic images, a very visible illusion-booster. The speed also enabled us to handle models with lots of detail within a visible cell, such as the Brooks House model. Figure 4 shows the two generations of systems built on PxPI5.

In this generation we saw the first textures, although speed slowed to 1 ups, since they were programmed

without microcode assist. Prof. Prins also devised a better scheme for partitioning, but it was not incorporated before it was overtaken by the Berkeley work.

3.6 Walkthrough 5.2 - Textures and the Ceiling Tracker

This generation, first demonstrated at SIGGRAPH '91, was dramatically different by the use of texturing at full speed. We also began to use Walkthrough with the new UNC ceiling tracker, which allows a radically more natural walking about motion.

4. Four research fronts of virtual environments projects— steps and lessons

Any balanced virtual environment project demands advances in four orthogonal dimensions. We have followed a squeaking-wheel strategy of system evolution, turning our principal attention at any time to the dimension whose shortcomings seem then to impair the illusion of presence most severely, while maintaining some modest on-going research in each dimension. Figure 1 represents these dimensions. We discuss each dimension separately, explaining for each the milestones in our system evolution.

4.1 Faster

For a dynamic Walkthrough experience, the illusion of motion is crucial. Good motion with almost any polygonal image that has perspective and obscuration will be perceived as a walkthrough of some validity. Poor motion is an illusion killer, no matter how rich the visual image.

Update rate. Our prototype, Walkthrough-I, operated at one update per three seconds. There is no illusion of continuous motion, but instead a series of snapshots. Some of our early work with molecular models showed, to our amazement, that when rotating a molecule, 1 ups yields an illusion of motion and kinetic depth effect, although one does not begin to get the illusion of *continuous* motion until 7 ups is reached. Edison's work with silent movies showed that general scenes gave a passable illusion of continuous motion at 12 ups, if each update is displayed twice with a short darkness interval between. We all remember the jerkiness of those silent movies, however. The experience of MacDonald-Douglas Aviation is that at 16 ups seasoned pilots find flight simulators obviously jerky; at 18 ups they detect that something is wrong but cannot identify what, and that at 20 ups they find flight simulators to move realistically. Our experience shows that for walking through architectural models, 15 ups seems a satisfactory lower limit. We routinely operate at 22-27 and find that good. We can see the difference between 15, 18, and 25.

Lag. The end-to-end system delay, perceived as the interval between a user's motion and a resulting change in the visual image presented to him, is a much more demanding parameter than is update rate. Moreover, it is radically more demanding when the user moves about among objects that should *stay still in space* than when he looks at the same world through a screen, navigating with a joystick.

We measure lags between 125 and 250 milliseconds. Much work in the virtual environments project is directed toward lowering the lag. We do not know how low it needs to be not to be noticeable, either from our own observations or from the literature

Computational speed as a system enabler. Better-looking visual images require techniques such as anti-aliasing, Phong shading, texturing, all of which require more computation per image. Stereo viewing doubles the computation per update. Big, detailed models require the rendering of lots of polygons: radiosity

requires rendering perhaps 10 patches per polygon. Dynamic, movable objects in the scene require both dynamic model editing and the continual testing of hand position against bounding boxes around each movable object.

In short, computational power provides the limit on how much model and how much fidelity can be achieved at any prescribed level of motion realism. We have continually had more effective computational power provided, by hardware and by algorithm, after we first reached the 15 ups threshold, and we have taken each increment and mostly used it to provide more visual and model realism. The Pixel-Planes 5 machine peaks at 2 million Phong-shaded polygons per second; we use it all and need more. The PixelFlow machine is projected at 40-100 million polygons per second; we expect to use all that.

This is one of the reasons for continuing work on the Walkthrough application – of all the applications in the UNC laboratory, it most thoroughly stresses the high-performance graphics engines developed here.

Model partitioning. The average depth complexity of our Sitterson Hall model is on the order of 5, so most polygons are concealed most of the time, even before one takes direction of gaze into account. Some of this concealment is trivial to predict, except in staircases, etc., only one floor's contingent of rooms can be seen at a time from within the building.

An obvious technique for speeding the computation of architectural walkthroughs is to partition the model into cells and at any time to render only those cells that are potentially visible from the viewpoint. Manually partitioning our Sitterson Hall model gave us approximately a 5-fold speed increment, at the cost of great human labor. Later, Airey devised automatic partitioning algorithms that gave most of this improvement, a factor of 4, with elaborate pre-processing but without manual labor. Sitterson Hall turned out to partition naturally into 269 cells.

Faculty member Jan Prins sketched a run-time algorithm using rooms, known open portals in walls, and direction of gaze that should give substantially better dynamic performance by evaluating a more precisely bounded set of potentially visible cells. We were unable to follow up on this work before Teller and Séquin at Berkeley independently developed the same approach, and together with Funkhouser generalized it to 3-D.

4.2 Prettier

Visual fidelity is of real importance when one uses an architectural walkthrough to judge the esthetic satisfactoriness, as opposed to just the workability, of a new design. (Workability can be judged with surprisingly coarse 3-D mockups and coarse 3-D visualizations in a way that it cannot be judged from 2-D floorplans and elevations.) We have continually worked to enhance the sensory fidelity of our images, not only visual, but also aural. Several milestones have each been noticeable.

Directional illumination. Our first efforts used flat shading, and we found this quite crude. The effect is essentially that of walls and floors in Saturday-morning TV cartoons. Even in Walkthrough I, our prototype, we therefore moved to an illumination model that provided a directional diffuse illumination combined with a flat ambient light. The direction of illumination was changeable to represent the annual and diurnal position of the sun, and we ignored the fact that roofs and ceilings intervened between our walls and the putative source of the directional diffuse illumination. Even so, the directional illumination noticeably helped.

Anti-aliasing. The jaggies are an illusion breaker, and architectural models show strong perspective effects and hence lots of diagonals. The PxPI4 team added microcoded anti-aliasing by supersampling, accomplished

by jittering the center of each pixel and recomputing the whole scene, blending the resulting images. This method was especially suitable for that machine, since the 256K processors do not intercommunicate.

Image sweetening. We wanted to have the effect of anti-aliasing before we could accomplish the real-time computation necessary. We therefore used the technique, often important in real-time graphics, of calculating a degraded image while the viewpoint is moving, and then sweetening the image while it stops. With anti-aliasing, the image continually improves as each sample is averaged in; the entire process takes about a second. The improvement in fidelity due to anti-aliasing was striking; the jaggies in unsweetened images while one was moving was discernable but minimal.

Radiosity illumination model. We then moved to calculate illumination according to the radiosity model for diffuse illumination pioneered by Greenberg and colleagues. This required hours of precomputation on the model, and it required displaying patches that turned out to be approximately one-tenth the size of our building polygons. In other words, the real-time display task is multiplied by ten. The effect is immense, the illusion of presence is radically improved.

Here, too, we wanted radiosity illumination before we could afford the run-time computation, so we used sweetening. At first, we sweetened the image as fast as we could, but this proved to be too fast for so drastic a change in image – the discontinuity broke the illusion of presence. We therefore interpolated between the crude image and the sweetened one, with the number of interpolants and thus the total sweetening interval as a run-time parameter. We found 60 interplating steps to be good, thus about a 4 second interval. The effect with radiositized images was dramatic – as soon as the viewer stopped moving, the shadows dynamically fled away to the corners of the room, leaving a properly radiositized view.

Radiosity and its adaptive calculation has provided many interesting research problems, including one of those treated in Airey's Ph.D. dissertation. The problem of breaking a polygonal model into mutually exclusive and exhaustive radiosity patches which do not create illumination discontinuity artifacts was solved in Nimscheck's M.S. thesis.

Visual fidelity tests. We built the virtual building models before the real buildings were built. This gave us the opportunity to videotape the real buildings after they were constructed, and to compare the videotaped interior appearance with that generated synthetically from the model. We have done this for Sitterson and for Orange Church. The process is underway for the Brooks house.

Nighttime and daytime models. We can explore a building at night, using the radiosity illumination from the building's own light sources, or in the daytime, using radiositized diffuse illumination from the bright sky as well as from interior lights.

Nighttime models look pretty good, both to the eye and in the visual fidelity comparisons with real views. The chief obvious shortcoming in the nighttime models is the lack of dim model reflections in the windows. We have built mirror code, but it is not yet integrated into our system. In a room such as the Sitterson lobby that has glass on four sides, the simulation of first-order reflections will multiply the number of potentially visible polygons by a factor of up to 5, thereby increasing computational load.

Daytime models do not yet look realistic. Several effects seem to account for this. First, when one is inside a building, the eye is naturally drawn to any spots of light, however small, including especially windows that depict the natural world outside. Our exterior modeling is so inferior to the everywhere-dense detail of nature that any view out a virtual window compares very poorly against videotape of a view out a real

window. Second, our exterior radiositizing still has a lot of artifacts, apparently reflecting poor patchification. Third, and we do not know how important this is, the dynamic range of our displays is orders of magnitude lower than the real-world dynamic range between the illumination inside a building and that outside.

Stereopsis. With the movement to a head-mounted display came a strong demand for stereopsis. Stereo requires the calculation of two images per update, burning half the computation capacity of the renderers; so on PxD14 we could only use the head-mounted display in monoptical mode. The extra speed of PxD15 made true stereo possible.

Stereo noticeably adds to the illusion of presence, but it makes much less difference in architectural modeling than in most applications. Architectural models have lots of parallel lines and planes, and lots of planes known to be perpendicular. Perspective is therefore unusually strong as a depth cue in this application; indeed, we considered its calculation essential even in our most primitive prototype. Obscuration, perspective, and the kinetic depth effect provide so strong a set of cues that stereo is almost frosting.

Textures. PxD15's capability of providing real-time calculation of anti-aliased procedural textures made an immense difference in visual fidelity, and we have not yet exhausted the power this capability gives us. The level of visual realism added by each texture is indisputably as strong as that provided by 100 more polygons/update, and we estimate it to be stronger than an additional 1000 polygons/ update, or 20,000 polygons/second.

An analysis of our house model showed that we would need at least 64 textures to do it justice, with 128 preferred. These numbers jolted the PxD15 team, who had originally provided for only 16, expecting that to be quite enough. Upon request, they brought that up to 32. We are currently using 29. The most used textures in our house model are woodgrains, acoustic ceiling, linoleum, wallpaper, and brick. This is, of course, no surprise; if we could have only two textures, we would choose woodgrains, then brick. More surprising are the use of textures for flames and for framed art. We need a texture for areas of books, and bump-mapped textures for trim moldings. We expect textures to have a dramatic effect on the yard. At this time the trees and shrubs are coarsely modeled with polygons.

Fire. Varshney simulated an open fire in a fireplace by using a small collection of translucent polygons, which are turned on in a pseudorandom sequence of combinations, about one combination per second(?). The flames look pretty good. When the radiosity calculation is run, it is run for each flame polygon separately, and at run time the results are combined by superposition to yield the fire's illumination of the room. The flame polygons are made bigger/brighter for the radiosity calculation than they are for the display. When one is in the room with the fire going and the lights out, the flickering illumination on the walls and ceiling is very effective, much more so than the fire itself. Needless to say, such a fire burns a lot of computation cycles.

Sounds. Viewing a 1989 videotape by Kotto and Petrisiitonen, Helsinki University School of Architecture, showing a projected development in Helsinki awoke us to the powerful illusion enhancement provided by even a limited set of sound cues in an architectural simulation. We consequently added monaural sound cues that are triggered and volume controlled by proximity to and entering the room in which the virtual sound source is located. Sounds used to good effect so far include wild birds singing, piano music, a radio talking, a printer running, a phone ringing, a toilet flushing, food cooking.

4.3 Handier

We have above sketched the ideal user interface. In no other part of our project have our views as to what approximations to the ideal are (1) tolerable, (2) good, and (3) quite satisfactory so changed with experience.

A Walkthrough interface for a research system must provide for two users, the one experiencing the Walkthrough and the one varying parameters to study system effectiveness and to learn user behavior. The former must be located at the walker, and move as he moves. The latter can be on a workstation screen, separate from the screen seen by the user. We will deal only with the walker's interface.

Output from the system to the user is the illusion-delivery mechanism. The additional interface capacity is that for input from the user to the system, with confirmation of receipt. We see the walker as needing the following input tools:

- a way of specifying viewpoint translation and orientation
- a navigation map to show where he is and his field of view
- a way of specifying where his hands are, at least relative to the viewpoint
- a way of specifying hand grabbing and releasing, with feedback.

Joysticks. We have at various times used velocity-modulating joysticks to specify translation and rotation of the user's head, hence his viewpoint. The user essentially flies through the model. Assume a building coordinate system in which z and x define the plane of the floors and are aligned with the major axes of the building. Up is then $+y$. There are six degrees of freedom; the natural thing is to put the translations on one 3-D joystick and the rotations on another.

This turns out not to be a good idea. One normally moves through a building doing translations in only two dimensions, x and z , and rotation in only one, about the y axis. Exceptionally one squats or stoops, or artificially changes eye height to get another perspective. Only rarely does one roll the head; one looks up or down, rotating about the x axis, somewhat more often.

Moreover, velocity-modulated specification of all three head rotations is troublesome, since one easily gets into orientations from which it is difficult to recover. A reset to level-headedness is indispensable with such a control. The twist axis of a joystick used for y translation is not intuitive. Is clockwise up or down?

We find the most natural mapping to two 3-D joysticks put the common motions on one, the uncommon ones on the other. This also facilitates one-handed operation. For the prime joystick, we use forward-backward for level motion along the direction of gaze, sideways for level motion perpendicular to the direction of gaze, and twist for head yaw. If the joystick allows 360° twist, positional mapping is better than velocity. On the secondary joystick, we map forward-backward to head pitch, sideways to head roll, and twist, yes unnaturally, to translation in y , following a left-handed screw – clockwise is up. We plan to reverse that latter, but don't expect it to help much.

The eyeball. We embedded a Polhemus 3-Space tracker in a pool ball with its bottom quarter sawed off. This proved to be a very natural input device, christened the *eyeball*. Translation in x and z , and rotation about y , are easily specified with the eyeball resting on a desktop or pedestal. Eye height above the floor is defaulted to a console-specified value; head roll and pitch are defaulted to zero. Yet the user can easily roll or pitch his head, and easily return to level. He can easily raise his eye-height, and a clutch button on the eyeball allows him to change the default, as when moving from one floor of a building to another. This is the best input device we have found for through-the-screen walkthroughs; it is not suitable for head-mounted display use.

Steerable treadmill. At the suggestion of James Lipscomb, we made a navigation device especially for walkthroughs, designed to allow the user to experience the scale of a space by actually walking it. We modified a non-motorized exercise treadmill by adding bicycle handlebars and sensors to determine the steering direction and treadmill rotation. This was used in conjunction with both the big-screen display and the head-mounted display.

As implemented, our treadmill was rather stiff, and no amount of adjustment or modification could make it really easy for walking. As a consequence, the average user's stride was $2/3$ of normal, so we moved the viewpoint forward $3/2$ times the measured stride, so that the number of steps required to walk a room would be true. The steerable treadmill provided quite a realistic walking experience, and it neatly solved the problem of the limited range of the head sensor on the head-mounted display. Nevertheless, it proved to be too slow a tool for exploring extensive models. The user wore out with the exercise and grew frustrated at the slow pace. The flying metaphors proved more useful for this kind of rapid survey.

Big screen (wide-angle) display. A normal workstation screen subtends quite a small solid angle (~ 0.6 steradian), so through-the-window walkthroughs inevitably have a certain detachment and remoteness. One is looking *at* a scene, not looking about *inside* an environment. Although we have developed four generations of head-mounted displays in our UNC laboratory, all of these have two problems that persist: low resolution and relatively narrow field of view. We therefore installed a rear-mounted video projector shining on a 4'x6' translucent screen, one which preserves polarization for stereo. The user can move himself, and his joysticks or treadmill, as close to the image as he chooses to focus, thereby getting a wide field of view (~ 2.0 steradians) and high resolution as well. The combination of the big screen and the treadmill works fairly well; given the fixed screen, it is natural to constrain motion to be in the direction of gaze, and that is natural with the treadmill.

Head-mounted display. A head-mounted display provides total immersion in a scene and the associated head tracking allows a very natural viewpoint change by head rotation and limited head translation. A variety of widgets in the virtual space allow the user to fly in a specified direction, shrink the world, stretch the world, and grab objects. When the head-mounted display is used in conjunction with the treadmill, the user can walk in one direction and simultaneously look about him in all directions.

Active floorplan. As a navigation aid, we provided with our prototype system a second screen which showed the floorplan of the floor on which the user was walking, and an indication of his height (as well as parametric controls, such as those for field of view and direction and mix of illumination). A cursor on the floorplan faithfully reflected the user's position and field of view.

This aid proved very valuable. We found that most users in fact did map navigation, moving to a desired viewpoint and waiting for the scene to paint, which took 0.33 seconds. Unfortunately, this feature has not been preserved in any of our later systems.

Map navigation versus scene navigation. We observed that as soon as our update rate began to approximate continuous motion, i.e., upon the transition to 3 ups, users seemed comfortable navigating by the familiarity of the scene, rather than by the floorplan. Even so, people easily get lost in Sitterson, in the model and in the real building. The symmetry is too high, and symmetric scenes are not sharply dissimilar. Indeed, we find people getting lost in our 11-room house model, although hardly ever in the house itself.

We are eager to study more about user navigation in comprehensive building models, especially when and how solidly users switch from map navigation to scene navigation.

Room-size ceiling tracker. The UNC head-mounted display project has developed a stereogrammetric head tracker, in which head-mounted lateral-effect diode cameras image some of 960 ceiling mounted beacons as they are flashed. The resulting system allows roaming and tracking over a 10' by 12' space, large enough to be interesting for architectural modeling. Although all the head-mounted apparatus makes an awkward weight, and the total system lag is still noticeable, the naturalness of the walking motion begins to approach our ideal.

Translation, but *not* rotation. The tracker is not large enough to encompass most rooms, so a means of moving the real 10'x12' space about in virtual space is necessary. After trying various schemes, we have settled on a pair of buttons, mounted on the hand or the belt, that allow one to fly the real space forward or backward in the direction of gaze in the virtual space. We have determined that many evils in confusion of navigation and in disorientation discomfort seem to arise as real space is rotated relative to virtual space, so we have made no provision for reorienting those coordinate systems relative to each other. North in real space is always north in the virtual building. This seems to aid navigation greatly. Notice that the steerable treadmill and joysticks did not have this property.

Hand tracking and grabbing. We have mounted a Polhemus source on the ceiling tracker headgear, and the user wears a Polhemus sensor on one or both hands, together with a grab button. This enables the user to open and close virtual doors, activate virtual light switches, pick up virtual objects such as the frying pan on the stove, and move some furniture around. Pressing the grab button while the hand icon is within the bounding box of a pickable object accomplishes picking. There is no feedback, except that the object begins to move with the hand. Objects set down on the counter or the floor behave properly; objects partly overhanging the counter edge do not. We use a highly simplified physics model.

Collisions, real and virtual, and the bounding ribbon. The ceiling tracker uses a ceiling supported at the corners by four 4" steel posts. Walkers using the head-mounted display are not able to maintain a real-space model that tells them when they are at the edge of the tracking range, or when they are about to collide with a post. Although it promised to be an illusion-breaker, we finally resorted to superimposing a 4"-high red virtual eye-height ribbon delimiting the active tracking area on top of the building model. This turns out not to impair the illusion significantly, and it radically helps the confidence of moving about blindfolded. One begins to think of the building model as having a real red ribbon at eye in it. Translation of the real space within the virtual space moves the four-sided ribbon in the virtual space.

With all of our systems through Walkthrough 5.1, the user could freely pass through walls, closed doors, etc. This discrepancy with the real world proved to be a substantial illusion-breaker. In the latest system we have added the detection of collisions with virtual walls, but not yet with virtual furnishings. Once one has detected a collision with a virtual wall, what does one do then? We cannot keep the user from walking on. What is scene fidelity in such a case?

We translate the wall, and indeed the entire real-space, if the user persists in "pushing" on a virtual wall after he has collided with it. Translation is in the direction normal to the wall, not the direction of user motion. It is as if the user had walked up against the red ribbon and pushed it, and its whole coordinate system, with him.

Illumination control. We provide for up to 24 independent lighting circuits in our building model. Each is controlled by a rheostat on the screen of the auxiliary control. Because the radiosity calculation is linear, superposition applies, and the lighting component on each patch due to each illuminant is separately stored. At

run time the contributions of each illuminant, scaled by its rheostat, are added together to yield the final brightness and color of the patch.

The system lag is noticeable in doing the computations. Lights come on about a second after the switch is thrown, whereas a 100-millisecond interval would appear simultaneous to the human viewer.

We plan to provide for activating lighting circuits by providing grabbable virtual switches on the walls in the building models.

User malaise. Most walkers using the Walkthrough 5.2 system have been in the virtual buildings only a few minutes, since they were part of a multi-party demonstration. We have had serious users who were studying building design issues and stayed in the virtual environment for a quarter hour or more. Under these conditions some users become very uncomfortable ("Get me out of here!"); some get motion sick. Others have no ill effects whatever other than weariness from the 2 kg weight of the headgear.

We believe the motion sickness arises primarily from the system lag and the consequent conflicts of vestibular and visual cues. We have also identified low blood sugar due to hunger as a contributing factor in at least two cases. Consequently we keep a pack of nabs taped to the computer as an emergency ration. The nabs noticeably helped the two cases of simulator queasiness recover.

4.4 Realer

One wants the simulation to approach closer and closer to the experience of the real thing. We see this aspect of simulation research as having two almost independent components. Besides faithful visual rendering (progressively overcoming the limitations of computer graphics), natural motion (progressively overcoming the limitations of discrete position sampling), and natural interface (progressively overcoming the interaction gulf between the user in the real world and the image in the virtual world), one needs progressive improvement in the *underlying reality* – a model that more and more closely approximates the real building.

Secondly, *research* in virtual environments requires that the simulation receive *real use*, in order to determine the aspects of the real-world object to be modeled in declining order of relevance and importance for some defined operational user task. This seems the only way to evaluate virtual environment systems.

4.4.1 Modeling.

A *model* is a synthetic construct that represents or duplicates a real or natural object, *in some, but not all*, of its properties. This definition seems to cover the full range of models, from a mathematical model of the law of gravity to a cardboard scale model of a house. Models are then defined by naming the object modeled, and by specifying which of the object's properties are included in the representation.

Properties of Walkthrough Architectural Models. Our Walkthrough models

- model a building, or some subset of it, and perhaps its site
- represent only the visible exterior and interior surfaces of the building and its furnishings, and certain audible sound sources. Invisible building aspects, including structure, mechanical systems, plumbing, and wiring, are not included in the models.
- represent to an accuracy of 1.0" the geometry, down to some chosen level of detail.
- represent visible surfaces with polygons that are colored and that may be textured and translucent.

In particular, curved surfaces are represented by polygonal approximations.

- represent surface colors with 256 levels each of red, blue, and green components.
- represent illuminants on a scale of brightness that is ordered, but not strictly linear or logarithmic.
- represent sounds -----

Models Built . We have built, and we currently maintain and are prepared to distribute three major models:

- **Sitterson Hall**, the home of the UNC Computer Science Department, O'Brien and Atkins, architects.

4 floors

314 rooms, counting toilets and storage rooms, but not closets

24,473 polygons

269 visibility cells

The lobby levels and adjoining rooms of Sitterson, only, is radiositized and can be distributed as a

separate model. It has

14 rooms

5,677 polygons

54 visibility cells

16,267 radiosity patches, each rendered as a separate Gouraud-shaded polygon.

- **The Fellowship Hall and Sunday School addition to Orange United Methodist Church**, Chapel Hill,

Böhm-NBBJ, architects.

2 floors

49 rooms

9,611 polygons

The Wesley Fellowship Hall itself, and other rooms on that level are radiositized and can be distributed as a separate model, containing

34 rooms

6,974 polygons

106 visibility cells

31,055 radiosity patches, each rendered as a separate Gouraud-shaded polygon.

- **The Brooks house with projected addition**, Chapel Hill, F.P.Brooks and Knox Tate, architects.

2 floors

23 rooms, counting halls, bathroom/toilets, storage rooms, but not closets or outdoors spaces

13,077 polygons

38 visibility cells

32,517 radiosity patches, each rendered as a separate Gouraud-shaded polygon.

We built but no longer maintain a less detailed model of Sitterson, and a model of an imaginary 6-room fully furnished apartment.

Model engineering. We have learned that models of over 5.000 separate elements, polygons in the case of

our building models, qualify as serious work products, requiring the same order of effort as do substantial computer programs. Just as one needs to bring special disciplines and tools to bear in the construction, updating, and maintenance of programs of over 5,000 lines of code, so one needs to bring special disciplines and tools to bear in the construction, updating, and maintenance of serious models. This has been one of the most important lessons of our experience. The whole subject of model engineering is detailed further in Appendix 1, a panel contribution given by Brooks at SIGGRAPH '91.

Model structure – hierarchical and flat – and model compilation. We increasingly find it useful to think of building models as hierarchical compositions. Figure (?) in Appendix (?) shows one possible hierarchical decomposition. In visibility partitioning, for example, entire floors make a logical first partitioning.

One may well consider the individual polygon to be the smallest contiguous area with a uniform surface color and texture. For rendering with a radiosity illumination model, this will need to be divided into patches, but there are often computation economies in recognizing the coherence shared by a polygon. Similarly, we identify *plates*, such as walls and floors as units for visibility calculations, even though a wall may be painted different colors in different parts, or have polygonal art on its surface.

It is interesting to note that an inherent duality arises in building models. For many purposes, such as our interior decorating, or indeed our viewing, it is useful to consider a building as being made up of mutually exclusive and collectively exhaustive volumes, namely rooms. For other purposes, such as construction, it is useful to think of the building as being subdivided into the walls, floors, and ceilings, rather than the spaces they enclose. In this way of thinking, a single plate separates the third floor from the fourth floor. Similarly, a single planar wall may separate a whole row of offices from a hall. During the design process, one rarely thinks of moving one office wall; one might well consider moving the entire wall for all the offices, and shifting the hall in width or location.

Although hierarchical models are convenient for design, for navigation, for comprehension of a building, and for saving computation in some algorithms; run-time rendering is usually fastest with a flat model consisting of the patches, each of which is rendered as a shaded polygon. This leads one to manage models as one does computer programs, keeping a *source* model, in hierarchical form, and a *run-time* model which is constructed from the source model by a *model compiler*.

A new theoretical structure for architectural models

Plates, e.g. floors

Partitions, e.g. walls

Portals, doors and windows

Polygons

Ports, holes in polygons

Patches, radiosity units

Model tools. Building a serious model by specifying more than 40,000 polygon vertices by hand is ridiculously tedious and error prone. We and other modelers therefore use model-making tools. Increasingly these are CAD systems designed primarily to help architects develop designs and working drawings.

For our first prototype, modeler Dana Smith recognized that buildings involve primarily rectangular spaces with constant separation from one floor to the next. He therefore took the Caesar tool developed at Berkeley

for the layout of rectangular components on VLSI chips and used it, together with a simple vertical extrusion operator, to create his 8,000 polygon model of Sitterson Hall.

Early designs for the Brooks house were done in MacDraw and later MacDraft, which included important features that substantially aided the architectural task..

AutoCAD and AutoCAD AEC. When we decided to move beyond this prototype to a regular research project, we standardized on AutoCAD as our principal building modeling tool, not because we like it best among architectural CAD tools, but because it is by far the most often used by architects, and we hoped to be able to import models more easily by using it. AutoCAD is pretty bad as an architectural modeling tool. It was originally developed for designers of mechanical parts, and it shows its ancestry in many awkwardnesses.

During the years we used AutoCAD, some of its other architectural users developed a package of macros known as AutoCAD AEC (for *architectural, engineering, and construction*). These helped considerably in avoiding and overcoming AutoCAD weaknesses.

MiniCad+ as a house design tool. Brooks decided that MiniCad+ by Graphsoft was the architectural design tool of choice on the Macintosh at the time he was designing his house extension, so his plans are all in that format. We tried to take DXF standard CAD interchange format from MiniCad+ into AutoCAD to get his plans into our system for model-building, but a variety of difficulties prevented that.

Virtus Walkthrough as a modeling tool. About two years ago, a start-up in Cary, N.C. developed and sold for the Macintosh a building walkthrough program in which one models buildings in terms of the rooms as individual volumes, rather than the floors and walls as structural elements.

We decided this is a much more natural and practical way to develop models, so we in 1991 entered into a collaboration with Virtus Corporation under which we β -test Virtus modeling software and offer feedback, and we render Virtus models on our radically more powerful visualization system. Even though we do not need the Virtus product's rendering capability, its modeling capability promises to be very useful to us, and we have standardized on it as the basic modeling tool for our next-generation models.

4.4.2 Real use

Sitterson balcony decision. The Walkthrough I system, whose construction was motivated by an argument concerning the lobby balcony was indeed used to settle that issue.

Church meeting review of walkthrough videotape. Walkthrough 4 and the fellowship hall model was not completed early enough to help the design process. It was used to make a videotape of a walk in the south front door of the Wesley Fellowship Hall, in a great arc through the south end, northward the entire length of the hall, and into the kitchen, around it, and back out, and then in some of the Sunday School rooms. This tape was shown at the church congregational meeting which was to decide whether to proceed with construction. They voted to do so.

After the meeting, the two architects repaired to our laboratory, for they had never seen the Walkthrough system. What was intended as a half-hour demonstration turned out to be a two-hour exploration of the space they had designed. They were very excited, and saw nothing they needed to change.

House look at music room. The Brooks house remodeling project and its Walkthrough model were done as Walkthrough 5 was being developed and brought up, and it served as the driving problem for that development, revealing many subtle bugs, such as an error in the aspect ratio of the visualization system. The most frequent tour during that period came in a window of the sun porch, into the music room, down to the fireplace, and then about. This is the tour that has been widely seen, including PBS television.

House design changed, team unable to model real, rather different plan. Partly as a consequence of seeing the music room in particular, and partly as a result of shifting design desiderata, the Brookses made substantial changes in the remodeling plan, saving a bedroom that had been scheduled for destruction and switching the spaces used for the music room and a study. This change was in September of 1990. The project team, however, was totally absorbed during the 1990-91 year in getting Walkthrough 5 running with the new ceiling tracker, and in polishing existing models for the UNC major display in the Virtual Realities Gallery at SIGGRAPH 91 in August of 91. So they have not yet been able to revise the Brooks house model to reflect the plan that was actually constructed during the 1991-92 year. Hardware *is* easier than software.

House look at kitchen. The Brooks remodeling is in fact divided into two phases, the first of which encompassed adding four rooms and reworking three, and the second of which plans a total reworking of the kitchen and the former den. The second phase has not yet been scheduled, so the team has built the updated model of the proposed kitchen and den.

This has proved to be useful in the design. One design issue was the width of a cut-through in the counter. The simulation shows the proposed width to be not only adequate, but also comfortable. It has led to a different furniture arrangement in the breakfast nook.

5. Painful Shortcomings and Future Work

5.1 Faster

Lag. The chief illusion-breaker on the speed front is lag, not update rate. We are seeing 100-200 ms., and it must get below 50. The HMD team has a major effort on end-to-end system lag; the Walkthrough project will show, and stress-test, their accomplishments.

What update rate is fast enough for realism? European television finds that 25 frames per second, 50 interlaced fields, suffices for motion realism. We should be able to operate quite satisfactory at 25 ups, or 50 stereo frames per second. If this does not seem fast enough, we will investigate interlacing left and right eye images, which should help, both in the HMD and on the projection screen.

How many polygons per second will we need to render? It's hard to tell. It is clear that texturing trades off against polygon rates, both in visual effect and in processor time. We are now seeing some 20,000 visible patches at peak periods, although the average is way below that. As we build more detailed models, we believe we will be regularly seeing 50,000 patches on the screen at once, of which perhaps half will be textured. At 50 frames/second, that calls for a polygon rate of 2.5 million/second, half textured., or 7.5 million/second with 5-fold supersampling for anti-aliasing. The PixelFlow machine now under construction promises to do this easily.

The theoretical limit of necessary speed for a single stereo image, of course, is one patch per pixel, but we do not believe that scenes even approaching that will have enough coherence to be conceptual comprehensible or pleasing. Note that if there are 50,000 polygons on a HDTV screen, an average one has about 30 pixels.

Progressive refinement. We have in the past used progressive refinement, or sweetening, to get high-fidelity visuals when we lacked the raw power to do so, first anti-aliasing, then radiosity. We may well need to go to it as we get more detailed models, so that the finer polygons are rendered when the viewpoint is still and most do not need updating.

Z-differentiation in model detail and texturing. Our laboratory, and others in the field, routinely use z-differentiation to manage high-polygon loads. Objects in the near field are approximated with many polygons and painted with elaborate textures, while those farther away are done with less detail and coarser textures. We shall surely need to do this for textures; we may want to do it for model detail.

5.2 Prettier

How far are we from realism? We are quite far away from fooling anybody visually. Therefore we expect to devote a fair proportion of near-future work to improving the visual image. As one looks at our comparative videotapes of real and modeled scenes, several things immediately leap out.

Detail. The most striking difference between our pictures and reality is the level of detail in our models. Except for the Bauhaus period, human beings have always used lots of detail for ornament. Carved wood or stone is the height of elegance in many periods. Even the restrained architecture of the neoclassical period uses elaborate trim to accent and offset plain expanses of wall. Today's simple home uses trim in many places, and few wall areas are left undecorated by either art or wallpaper, both of which add visual complexity. Our planned research on detail will be set forth under 5.4, Modeling.

Nighttime models need reflection mapping off all glass surfaces, certainly off vertical axial glass planes. Assuming no glass in floors and ceilings of buildings, this requires multiplying all our visibility cells by about 5. This will strain PxPI5 polygon-processing rates, and it will strain memory capacities.

Varshney has already built the reflection-mapping code, and we have used it in smaller models. We need to integrate it into our big models.

Daytime models need more accurate sky and foliage reflectance models. We have never produced a really satisfactory radiositized model of a building, even the house, in its site.

Yard models are currently very primitive. The best we have is for the house. It includes a 9-plane terrain model of the yard, which should get us quite a ways, but the trees, grass, and shrubbery are pretty bad. We need to use the remaining available textures to represent grass, leaves, and bark. We need to enhance the model so that tree trunks are approximated at least by two intersecting planes, if not by cylinders.

Lighting. We do not yet notice the absence of specular reflection, except from planar surfaces where reflection mapping will do the trick. Likewise, we are not yet conscious, when we examine comparative videotapes, of the effects of the approximation of uniform diffusion off all surfaces, independently of angle.

The single lighting effect that would make the most dramatic visual difference is to map sunshine through the windows onto floors and furnishings, and to propagate that light by radiosity. We want to experiment with this for a static sun. Bell is working on dynamic radiosity calculation. We do not expect to have it for the house model in Walkthrough 5 versions, but PixelFlow should make it possible.

Textures. There is much to be done to make our textures more realistic, as well as to incorporate more texturing information in our models. The woodgrain and brick textures are quite good; the acoustic ceiling texture is pretty good, but a little heavy-handed, and the linoleum and wallpaper textures need work, mostly in the direction of making them more subtle.

We have not yet used the Phong-shading capabilities of PxPI5, because we have mostly planar surfaces. An interesting experiment would be to do moldings by bump-mapping textures.

Curvature. Pixel-Planes 5 incorporates quadratic primitives. We have not yet used them, but approximate curved surfaces with shaded polygons. Almost all the curves in our models are quadratic. We shall start by using cylindrical surfaces for columns, and for moldings.

Directional sound. Our present sound cues are localized by cell, and volume modulated by the viewer's closeness to and presence in the cell, but they are not directional. We have acquired hardware and software for the Macintosh that will enable us to produce two directional sound cues (azimuth only) plus one non-directional background sound. We shall experiment with seeing the effect of this on the total illusion.

5.3 Handier

How far are we from natural motion and manipulation? In one sense, we have made immense progress in the user interface, and by many experiments we have arrived at what we expect to be our standard

techniques for handling some problems.

In another sense, one may feel that he is moving about a realistic space in a realistic fashion for short intervals, but one frequently encounters illusion-breakers. There is much to be done to make the naturalness of the interface, and hence the illusion, robust.

In yet another sense, the task is essentially difficult. Although we have done, and are now doing, extensive work in force display, it is not easy to see how to make force output readily wearable, so that one feels the virtual wall when he runs into it.

Navigation. The wide-area ceiling tracker is a major contributor to illusion realism, even though it does not cover very much a real building at a time. One really moves about, relatively untethered, in a space large enough to feel like a room, even though it is only part of one.

Ceiling tracker futures – big ceiling. The HMD project has a major effort on improving the ceiling tracker on many fronts. The construction of a large ceiling, big enough to more than encompass a room, will make the most difference to the Walkthrough illusion. The 12'x10' space is just not enough. More seriously, when one is near the edge, say 2', one or more cameras go off the ceiling, and the update rate drops radically. If one looks down, say at the stove or sink, a camera is apt to go off ceiling. If one squats down to look up under things, a camera is apt to go off ceiling.

Headgear weight. At present the headgear is so heavy as to be a fatigue factor, especially on small-stature viewers. Efforts to lighten it may make an important difference during long user sessions.

The treadmill, a bicycle, and unconstrained flying. We have also rejected unconstrained flying inside a building as a normal mode of exploration. We feel it is important to maintain the real and virtual world coordinate systems in rotational registration, although we find translation of the real-world space within the virtual one to be relatively natural and not disorienting.

We have retired the treadmill. It is disorienting and it is just too much work. We may yet experiment with a bicycle in large building models, since we have an instrumented bicycle and it is a low-cost experiment. This would be our last remaining effort at vehicle-metaphor exploration.

Landmark ribbons of real world and wall collision behavior. We believe we have arrived at suitable standard techniques for these problems.

Secondary view. We need to experiment with secondary views, in particular a floor plan, put in a lower corner of the screen.

Elevators, escalators, and stairs. Stairs are hard to simulate. We have tried using Warren Robinett's escalator that starts moving as soon as one is within an entry bounding volume at the top or the bottom, but it didn't seem especially natural or robust. Elevators work fine. So does push-button constant-velocity vertical translation, but one needs a detent at normal eye height above the floor.

Tactor for feedback when picking objects. Our current arrangement uses a pushbutton on a glove to indicate when one wants to grasp a movable object. We have previously used a guitar pick mounted on one finger with a Polhemus sensor and a pushbutton. It is more convenient and more natural than the glove. We are acquiring two small shape-memory metal tactors that will impulse the finger when energized. We will try

this as a sensory feedback cue for grasping. We have previously experimented with a pager buzzer. It's better than no feedback, but not at all natural.

5.4 Realer

Real use. The house model will get more real use in the refinement of the Brooks kitchen design. No other real use of the system is planned.

Our system has reached the quality where it could be useful to real architects. The natural next step in this research would be for us to enter into active collaborations with one or more architectural teams. All the experience in our laboratory has been that outside collaborators, because they are outside, add a great deal to this kind of research. They bring real applications that raise unanticipated problems, stimulating new research directions. . They rank quality components in different ways from what we expect. They ask for capabilities we never thought of. They make unexpected kinds of uses of our systems.

In spite of all this immense value of outside collaborations, we have made a decision not to prosecute this research to that proper next step. We have so many important research opportunities that some hard choices have to be made, and Brooks has concluded that his efforts can cover collaborations with chemists or with architects, but not both, and he has chosen to continue his long-term work with chemists. (Collaborations take P.I. time for tasks that can be delegated only to seasoned faculty.) Hence we will continue to develop the system with internal users only, as a stressor and application for our technology work in virtual environments.

Model engineering. We noted above that lack of detail is a highly visible difference between our present illusion and reality. How much detail is enough in architectural modeling? We plan to try to find out by maintaining and embellishing a single model, our house model, over a period of years.

We will use the Virtus modeler to prepare a much more highly detailed model of the Brooks house. We will attempt accurate colors, precise quadratic curves, refined textures, and architectural detail such as moldings that have not commonly been in building models.

Illusion effectiveness ranking. Heretofore we have used our own judgment to estimate the worst remaining illusion-breaker, so as to attack it next. In the coming years we propose to set up comparative evaluation of our illusions, not merely with reality, but with each other, using panels of lay judges to compare the total illusion using one refinement (more model detail, for example, baseboards and crown moldings) against another (better lighting, for example, sunbeams). It is especially important to compare textures with raw polygonal detail. We hope over a period of years to learn substantially more about what the human notices and cares about in architectural simulation.

5.5 Next-generation Walkthrough 5

Figure 5 shows the next-generation Walkthrough 5 as we now contemplate it, with the new or added characteristics shown under each of our four progress headings. The target date for this system would be August, 1993, if we have enough interested students and can fund them.

6. Technology Transfer

6.1 Stardent Use of Church Model

Stardent subcontracted with several of our project's graduate students to do a special project enhancing the church fellowship hall radiositized model for them to use in demonstrating the rendering power of their engine. In the event, it turned out to have too many patches to give reasonable dynamic walkthrough performance, so they used the model to make stills, which they extensively featured in advertising.

6.2 SGI work on radiosity.

Daniel Baum at Silicon Graphics Inc. took our model of the Brooks House project and used their own new radiosity techniques. Their work helped us find various defects in the model. They were able to do a rather better job at radiositizing it than our program did. They have used that model and the church model extensively in demonstrating their equipment, including on the exhibit floor at SIGGRAPH 90.

6.3 Berkeley work on partitioning.

Carlo Sequin and Seth Teller were stimulated by Airey's work on partitioning to devise their own approach to the problem, originally published in SIGGRAPH 91 as restricted to rectilinear geometries and two dimensions, but then demonstrated at the 1992 Real Time Interactive Graphics workshop fully extended to 3-D and able to partition in real time as the user walks. They have offered to make their work accessible to us. It is very pretty; we shall use it.

6.4 Virtus Modeling

As described above, we have entered into an active collaboration with Virtus to use and β -test their 3-D volume-based modeling package.

7. Acknowledgements

ONR. During 1987-88 the UNC department received a substantial 5-year University Research Initiative grant from the Office of Naval Research. The ONR people were very interested in Walkthrough and encouraged us to increase the team of graduate students using the ONR-URI funds. This enabled us to operate with 5-6 graduate students instead of the originally funded 3 for all of the NSF grant period.

PxPI team. Much of our progress is due to the outstanding work of the Pixel-Planes Project, under the direction of Henry Fuchs and John Poulton. Their development of the Pixel-Planes 4 and the Pixel-Planes 5 graphics engines, combined with our algorithm work, made it possible for us to achieve real-time motion, and then gave cycles that we could trade for various forms of image realism, model size, interface power.

PxPI grant DARPA and NSF. The development of the Pixel-Planes graphics engines has been jointly funded by the NSF Program in the Microelectronic Information Processing Systems Division and DARPA Information Systems Technology Office, under NSF Grant MIP-9000894.

HMD grant DARPA. Since the NSF and then the ONR grants ended, the more modest on-going research in Walkthrough has been funded as part of Head-Mounted Display project, Henry Fuchs and Fred Brooks, P.I.'s. funded by Defense Advanced Research Projects Agency Grant DAEA 18-90-C-0044.

NIH NCRR. The initial exploratory work on Walkthrough, leading up to the original NSF proposal, was funded by the NIH National Center for Research Resources under Grant RR-02170-03.

Chronological Bibliography (include all internal documents)

Brooks, Jr., F.P. [1986]. "Walkthrough – a dynamic graphics system for simulating virtual buildings." *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, S. Pizer and F. Crow, eds. UNC-Chapel Hill, October 1986, pp. 9-21.

Rohlf, J. [1988]. "Walkthrough illumination design document." Feb. 20, Internal Project Report. 12 pp.

Brooks, Jr., F.P. [1988]. "First annual technical report, Walkthrough project." Univ. of N.C. Dept. of Computer Science Technical Report TR88-035, plus VHS videotape.

Airey, J.M., and F.P. Brooks, Jr. [1989]. "Walkthrough – exploring virtual worlds; second annual report." Univ. of N.C. Dept. of Computer Science Technical Report TR89-012, plus VHS videotape.

Brooks, Jr., F.P. [1989]. "UNC Walkthrough project." *SIGGRAPH '89 Course # 29 Notes, Implementing and Interacting with Real-Time Microworlds*. pp. 1-1 to 1-4, 4-1 to 4-9.

Airey, J.M., J.H. Rohlf, and F.P. Brooks, Jr. [1990]. "Towards image realism with interactive update rates in complex virtual building environments." *Computer Graphics (Proc. 1990 Symposium on Interactive 3D Graphics)*, 24, 2, 41-50. **Won symposium best paper award.**

Airey, J.M. [1990]. *Increasing Update Rates in the Building Walkthrough System with Automatic Model-Space Subdivision and Potentially Visible Set Calculations*, Ph.D. dissertation, UNC-Chapel Hill. (Available from University Microfilms.)

Brooks, Jr., F.P. [1990]. "Interview with demonstration excerpts of Walkthrough." *SIGGRAPH Video Review*, 60, (August).

Brooks, Jr., F.P. [1991]. "Modeling – today's biggest graphics problem." Panel: "Computer graphics: more unsolved problems," *SIGGRAPH '91*

Varshney, Amitabh [1991]. "Parallel radiosity techniques for mesh-connected SIMD computers." Technical Report TR91-028, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, July 1991.

Brooks, Jr., F.P. [1991]. Testimony before the U.S. Senate, Subcommittee on Science, Technology, and Space of the Committee on Commerce, Science, and Transportation. (Included videotape of demonstration). May 8, pp. 11-23.

Teller, S.J., and C.H. Séquin [1991]. "Visibility preprocessing for interactive walkthroughs." *Computer Graphics (Proc. SIGGRAPH '91)*, 25, 4, 61-69. This paper, not done as part of this project, nevertheless resulted indirectly from it – the line of research was stimulated by Airey, Rohlf & Brooks [1990] and Airey's dissertation.

Alspaugh, J.C. [1992]. "A short guide to AutoCAD drawing primitives for 3D computer graphics models and the Walkthrough AutoCAD-to-polygon conversion program." Univ. of N.C. Dept. of Computer Science Technical Report TR92-019.

Funkhouser, T.A., C.H.Séquin, and Teller, S.J. [1992]. "Management of large amounts of data in interactive building walkthroughs." *Computer Graphics (Special Issue on the 1992 Symposium on Interactive 3D Graphics)*. 11-20. This paper, like Teller [1991], was not done as part of this project, but nevertheless resulted indirectly from it

Rhoades, John, G. Turk, A. Bell, A. State, U. Neumann, and Amitabh Varshney, "Real-time procedural textures," *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, Cambridge, MA, March 1992, 95-100.

Varshney, Amitabh and Jan F. Prins [1992]. "An environment-projection approach to radiosity for mesh-connected computers," *Proceedings of the Third Eurographics Workshop on Rendering*, Bristol, United Kingdom, May 1992, 271-281. (also accepted to appear in the Eurographic Seminar Series of Springer-Verlag later this year.)