Immersive Real-Time Audio/Visual Architectural Simulations

Steven Landau
Visiting Faculty, University of Arizona School of Architecture

Marc Grassi
Masters degree candidate, University of Arizona School of Architecture

Abstract
The authors have developed an approach to creating real-time audio-visual architectural simulations that allow any viewer to virtually explore the spatial and acoustical implications of a building’s design. Though somewhat limited in depth of the immersive experience, this investigation successfully demonstrated that architectural presentations can be augmented and enriched by incorporating sensory information other than the visual.

1 Introduction
Too often, architects overlook the value of senses other than the visual as a means of communicating their spatial imaginings. Our clients have grown to expect elaborate video and still imagery of as yet un-built spaces; however, supplemental sensory information can often enhance architectural presentations by creating more-convincing immersive simulations. In computer gaming, these ideas have taken root; we are familiar with the “first-person shooter” applications that permit real-time, user-navigated visual and audio movement through fantasy scenarios. Architects can quite easily adopt some of these techniques. By giving the viewer control over his or her position within an architectural space, and by adding dynamic audio information that changes in volume and pans stereo left to right in respect to the viewer’s instantaneous orientation, we can dramatically improve our ability to communicate the spatial and experiential implications of our designs. During the spring semester of 2002 at the University of Arizona School of Architecture, the authors of this paper developed and tested techniques for creating a new kind of architectural presentation tool. Through this investigation, we hope to have demonstrated the usefulness of digital media and CAD as a bridge between idea and artifact. Our techniques and findings are discussed here, accompanied by still images from the finished product.

2 Process
Fly-throughs and other (exclusively visual) scripted animations provide a limited and passive experience for viewers; the animator determines the path along which a camera is set in motion its speed along the route and the camera’s rotation in respect to the x, y and z axes. Watching such a fly-through, regardless of the sophistication of its production, is like watching any prerecorded video: communication is partial and one-way. The viewer, perhaps subconsciously, assumes that the designer has selected only the best angles and features to reveal, and that the real experience of being in the space would probably be entirely different.

Now, thanks to newly available authoring tools, it is reasonably easy for an architectural designer to construct real-time video and audio simulations that can be packaged as stand-alone executable programs, or that can be distributed via the Internet. For the effort described here, we chose to work with Macromedia Director 8.5 Shockwave Studio. Prior to this release, Director had no built-in 3D capabilities (although some fairly primitive “Xtras” were created by third parties that prefigured the features that now come as part of the software). The new version allows for the creation and
manipulation of 3D “worlds” based on the paradigm established for modeling and rendering applications like 3D Studio Max: any number of lights, cameras, and shaders can be added to a virtual environment, which also can contain volumetric models. The models can be generated through Lingo code, in which each solid or surface is given size and position properties, and then shaders, with preset ambient, diffuse and specular characteristics are assigned to each primitive. This process leads to the creation of highly compact and efficient models. But it is often cumbersome and counter-intuitive for architects and designers, who are accustomed to working in graphical environments like 3D Studio or AutoCAD. The alternative method is to create buildings, or parts of buildings, in another modeler, and then import them to Director through the .3dw format. Macromedia provides a plug-in for this purpose. Once the model is imported, Director assigns numbers to each 3D entity, which can then be called and manipulated with code in event handlers. Models imported in this fashion must be kept as small as possible, because the computer processing implications of real-time interactivity preclude large, complicated models with lots of lights and bitmap textures.

For the purpose of the project described here, a model was built in AutoCAD, opened in 3D Studio Viz, and then exported to .3dw format. The resulting file was then imported to a Director movie cast, and then dragged onto the stage. A camera was created, and the up arrow key was assigned to move the camera forward; once set in motion, the camera would continue moving until the up arrow key was released, then it would begin to slow down. The down arrow acted as a “brake”, causing the camera to come to an abrupt halt. The right and left arrows were used to turn. If both the forward key and the right or left key were pressed together, the camera would move in a circular fashion. Collision detection was used to prevent the camera from moving through walls or furniture; if an obstacle was discovered within 10 feet of the viewing position, the camera decelerated until it became motionless 6 inches from the detected barrier. Then the viewer had to turn around and move off in a different direction. Figures 1 and 2 show views of the interface during a virtual tour of the building.

**Figure 1:** The interface showing the exterior of the building and the streetcar in motion.

**Figure 2:** The interface as it appears during a virtual tour of the building’s interior.
The next challenge was to add a collection of fixed-position and mobile sound sources, and to modulate the computer’s stereo output so as to produce a convincing spatial acoustic image as the user moved around the building’s exterior, through the doorway, and inside. Figure 3 shows a plan of the building; numbers 1 through 8 indicate the positions we chose for the sounds. Sound number 1 was a streetcar; by clicking the “train” button on the screen, the streetcar was (visually) set into motion and the accompanying sound was also propelled forward at the same velocity and direction. The rest of the sounds were stationary: their audio characteristics only changed in reaction to the listener’s changing position.

In order to modulate the sounds in a convincing way, it was necessary to create a script in *Director* that repeatedly calculated the relative positions of each of the sounds vis-à-vis the listener. This sampling was carried out many times each second so as to eliminate disconcerting abrupt changes in sound characteristics. Once the instantaneous positions were established and saved as variables in the computer, a number of individual calculations were carried out, and the “volume” and “pan” settings for each of *Director’s* eight sound channels was adjusted accordingly. For each sound, we made a series of calculations. The first was to determine the distance between listener and source. This straightforward process is illustrated in figure 4. A sound’s loudness is calculated as the inverse of the square of the distance; so that every time the distance doubles, the volume of the sound is perceived as being reduced by 75%. This calculation is built into the script that adjusts the volume for each of the sounds many times each second:

\[
    \text{Intensity} = \frac{1}{\text{distance}^2}
\]
A second calculation was made to establish the “pan” for each sound, that is, the relative volume levels coming from the left and right speaker channels. This determination is affected by two factors: the azimuth of the sound source in relationship to the listener, and the swivel of the listener’s head in relationship to due north. In order to simplify this potentially cumbersome sequence of operations, we devised a simpler, yet more reality-driven solution. We placed one “ear” on either side of the listener’s “head”; these were two small boxes, exactly 10” apart, which moved and rotated with the head-cylinder, in a child-parent relationship. Then, by polling the distance from the sound source to each of the “ears”, we could deduce the precise angle of the “head” relative to a line radiating out from the sound source. For example, if

\[ \text{distance to right ear} - \text{distance to left ear} = 0 \]

then we know that the user is directly facing (or directly facing away) from the sound source. Given this condition, we set the stereo pan to “0” for that particular sound channel, which means balanced right and left speaker volume. If, on the other hand,

\[ \text{distance to right ear} - \text{distance to left ear} = 10 \]

then we know that the listener’s head is facing 90° to the right, and the volume to the left speaker is reduced to 0 for that channel\(^1\). For values between 10 and 0, the pan is set appropriately, to simulate the different intensities implied by head swiveling. Similarly, for a difference in distance between the ears of 0 to -10, we control the pan from balanced left and right to all left and no right.

Figure 4 (top): Diagram showing method for deriving distance, used to set sound intensity.

\(^1\) This is overly simplified: in reality, some amount of reflected sound always reaches the far ear; and the phase difference between sounds perceived by the ear facing the source and those reaching the one facing away from the source adds an extra spatial cue that enforces acoustical imaging.
Figure 5 (bottom): Diagram showing method for determining left-right pan and for distinguishing front-facing from rear-facing head position.

Once volume and pan have been established for each of the sounds in the virtual environment, one additional step must be taken before the final audio signal can be generated and outputted to the speakers or headphones. Earlier, we stated that if the distances between the right and left ears and the sound source were equal, then we could deduce that the listener is either directly facing or directly facing away from the source. However, these two conditions imply significant acoustical differences. The shape of the human ear, head and torso, enables us to distinguish between those sounds that come from in front of us from those coming from the rear. The physiological mechanism by means of which this mainly takes place is known as “pinna shadowing”; the pinna is the outer fleshy part of the ear that acts as a sound funnel, directing vibrating air in toward the bones on the inner ear where stimulation can be interpreted by the brain as sound (Blauret, J. 1996). We are subconsciously able to interpret the sounds we hear in the context of our personal experiences. For the simulation that is the subject of this experiment, attention was paid to achieving some degree of differentiation between front and rear sounds. This was accomplished by using the listener’s “nose” as a point to measure to and to compare with distances to the ears. We knew that the listener was within 90° plus or minus from directly facing the sound if:

\[
\frac{(\text{distance to right ear} + \text{distance to left ear})}{2} \geq \text{distance to nose}
\]

As each sound was being evaluated, if we determined that a sound was coming from behind the listener, we modulated the frequency of the sound as a means of simulating the pinna shadow. This technique met with mixed success, and many listeners were unable to distinguish the difference between front and rear sounds. However, given the interactive nature of the presentation, a user needed only to swivel his or her “head” using the appropriate arrow keys to cause the stereo balance to shift, and thereby obtain clues that could be used for deducing the actual direction of the sound.
Conclusion

Obviously, there are many acoustical effects that we have ignored, including reverberation based on the hardness of materials, specular reflection, echoes based on room geometry, occlusion caused by objects between listener and source, and Doppler effects in which the pitch of a sound shifts at the moment when the distance between the source and listener stops decreasing and begins to increase. Some new research, utilizing “beam-tracing”, has made progress in beginning to use the computer to generate real-time interactive acoustical walkthroughs that include many effects that serve to enrich the illusion of being in a real place (Funkhauser, T. 2000). This work is in the realm of computer science, and requires the use of very powerful computers; as of now it is not generally feasible to deploy these technologies for architectural presentations. However, as with computer animation, it is not unreasonable to assume that within a few years, tools for incorporating dynamic sounds in presentations will be affordable and within the grasp of most practitioners. The student project presented here shows an example of the kinds of “audialization” tools that might become available and quite straightforward to use. It is difficult to predict the ways that inexpensive, realistic environmental sound simulations could change the ways we think about architecture, but if the impact of visual computer modeling is any indication, we should expect this new technology for representation to lead to some level of recalibration in the ways we approach building design.

References