

Mark Hertensteiner (SUNet ID:hert)

My class partner and I each wanted to create our own image for the final project, so for my image I was responsible for all aspects of the project.

The Scene / Theme:

The image title is 'Vox Machina' (voice of the machine). It's theme centers on the idea of a machine, the robot, playing / making music on another machine, the synthesizer. The setting is a small stage performance (for an audience of machines?)

Assets:

I created the synthesizer, cables, goose-neck lamp, and sound wave objects (and the room / floor / wall, and out of frame cans to create spotlight effect from point light sources). The robot arm, table, Focal Speaker, M-Audio audio interface, and curtain were assets I found online from websites that included free3d.com and 3dsky.com. I also downloaded the seeds for the wallpaper and floor textures (the speaker and curtain textures were included with their respective models).

The synthesizer and lamp were both created in Maya. For the modular synthesizer I downloaded images of the individual module faceplates to use as textures maps. All other aspects of the synthesizer mesh and textures were created from scratch (based on actual synthesizer modules I own). The configuration and cable patching shown in the image is a fully accurate and plausible setup for electronic music making.

The sound wave mesh was created in Matlab, using actual samples from a digital audio (.wav) file. I essentially oversampled the audio file by a factor of 64 and projected it onto a complex sinusoid at the oversampled frequency. I then wrote the .obj file directly from Matlab with the proper formatting. The visible waveform represents approximately 0.01-0.02 seconds of audio.

Implementation:

In addition to the standard ray-tracer functionality (using the photon-map renderer), I implemented shadows through transmissive objects with attenuation based on transmissive object material color (though there isn't much opportunity to observe the colored attenuation in my image), point-light attenuation, and multi-threading based parallelization.

For multi-threading, I used the <thread> library, although I came close to successfully implementing a thread-pool (and may revisit this attempt as an exercise that could significantly further reduce render times - my 1080p 32 samples/pixel final image clocked in at about 14 hours).

I was also able to successfully troubleshoot object artifacts that were caused by shadow ray-casting by adding a normal-offset to the point from which rays to light sources were generated.

Remaining wishlist:

In addition to successfully implementing a thread-pool as described above, there are a few additional features I would like to have added to my project had time permitted:

1) LED light sources: currently I'm setting the ambient color for LEDs on the synthesizer to appear lit. I also created a version in which the LEDs were transmissive objects with a low-level light source in each one (and no ambient color), which was the original reason I implemented point light source attenuation (I eventually used it for the lamp light also). The rendering of this other version took an impractically long time to run (at 480x270 resolution and 2 pixels/sample, I didn't even let it run to completion, it was taking so long), which wasn't too surprising given the over 2 dozen lit LEDs in the scene. I still think, however, that this would have looked freaking awesome. I might have even reduced the intensity of other light sources in the scene to emphasize the led lights had it been feasible to do. I may revisit the project and create an LEDLight subclass of PointLight to have finer control of how these light sources are handled in iterated calls (perhaps only if a mesh intersection is within a certain distance of the light source will a ray be cast to that source) to counteract the additional rendering overhead.

2) Volumetric effect in sound wave: David suggested adding a low-level light source within the sound wave to emphasize it, which I tried, but it had little effect other than casting light over the entire scene. However, if were possible to place an attenuated light source within the sound wave that would light up just the sound wave mesh (like a partial fiber-optic effect), I think that would look pretty cool. I think this could be achieved with volumetric shading within the soundwave, which would be something else I would've liked to try given the time. Maybe scattering photons from an internal light source when they exit the mesh surface would be another way to accomplish a similar effect.

3) An alternative method for visualizing the sound in air that I considered attempting to implement was refracting rays as they passed through a spherical sound field, centered on the speaker by some index proportional to the individual audio samples at the distance corresponding to where the sound represented by each sample would be in space given the speed of sound. I abandoned this approach due to time and complexity constraints, both in the implementation as well as the anticipated render time, compounded by the fact that I had no guarantee whatsoever that the

resulting visual effect would be compelling in either way. Still I remain curious how this would look, and may revisit attempting to implement this method of sound visualization at a later date.

Conclusion:

Overall, I am generally pleased with the final image result. I accomplished nearly all of my goals for the project and successfully executed my original vision for my scene. I may render a 4k version on my own, but unfortunately given the time it took to render 1080p it wouldn't be finished before the Thursday deadline.