

Modeling the Milky Way Galaxy Data Set using 3D Audio

Nathan Dass

Thomas Jefferson High School for Science and Technology

July 31, 2013

Dr. Mark A. Livingston

Introduction

The main goal of this project was to write a program that would allow a user to easily process the vast amount of information given by large data sets by using a 5 monitor by 12 monitor panorama display which would be showing an interactive map of the Milky Way Galaxy. In order to allow the user to process the information provided by the Milky Way Galaxy, I used a software application called SoundScape3D, an application that provides an Application Programming Interface which allowed me to create sounds that appear to be playing in 3D space even when using regular headphones. I used SoundScape3D to play various sound files and have these sound files appear to originate from the location of an astronomical body that is being shown on the panorama display. Processing large data sets has always been a problem, but this project allows a user to look at 1236 astronomical bodies and almost naturally turn to the bodies that have qualities that stand out from the other bodies due to the various sounds being played at any given instance.

Body

The main components of this project are the 60 monitor display, the tracking system, SoundScape3D, the actual headphones, and the program that would communicate between all of the other components.



Figure 1. Part of the 60 monitor display with the Milky Way Galaxy being displayed

Since the 60 monitor display is controlled by an 18-node rack-mount machine cluster, the only way to communicate with what is going to be displayed is to talk to the cluster. In order to do this, I had to write a program which would create a socket and connect to the IP address of the cluster and its port number. Then, I was able to communicate with the cluster which allowed me to send and receive data about the state of the cluster. Furthermore, the locations of the astronomical bodies were represented as if they were in a sphere with one coordinate ranging from 0 to 360 and the other coordinate ranging from -90 to 90, so I had to convert these coordinates into 3D coordinates in order to give the effect of the sound originating from the actual position of the body on the display. The tracking system that was used is called InertiaCube3 by InterSense. Since the purpose of the sounds is to make the user change their head orientation to look at the different astronomical bodies, the apparent location of the sounds also needs to be adjusted by the same amount to continue giving the effect of having the sounds originating from the bodies themselves. Using the InertiaCube3, which was attached to the top of the headphones, I was able to figure out how much the user turned their head in terms of yaw pitch, and roll, which are the rotations about their respective principal axes, and then adjust the location of the sounds.



Figure 2. Picture of the InertiaCube3 which is then attached to the top of the headphones

SoundScape3D has various uses, one of which is to create sounds that play in 3D space. This software application not only allowed me to play any .wav file, but also allowed me to set the

location that the sound is being played, move the sound while the sound is being played, adjust the sound intensity, and remove the sound as a whole. In order to allow each component to communicate with each other, I wrote a program in Java that would first communicate with the cluster to find out what astronomical bodies are being displayed. Then, using the preprocessed data set, I would figure out which aspects of which bodies differed from the rest of the bodies. Using this information, I then communicated with SoundScape3D to either update the position of an existing sound or create a new sound for a new body that has appeared on the display. The program also constantly checks the orientation of the user's head, and modifies the sounds accordingly. However, since the SoundScape3D is in C++, I used an application called Uber App which would act as a middleman between the Java program that I wrote and the SoundScape3D API.

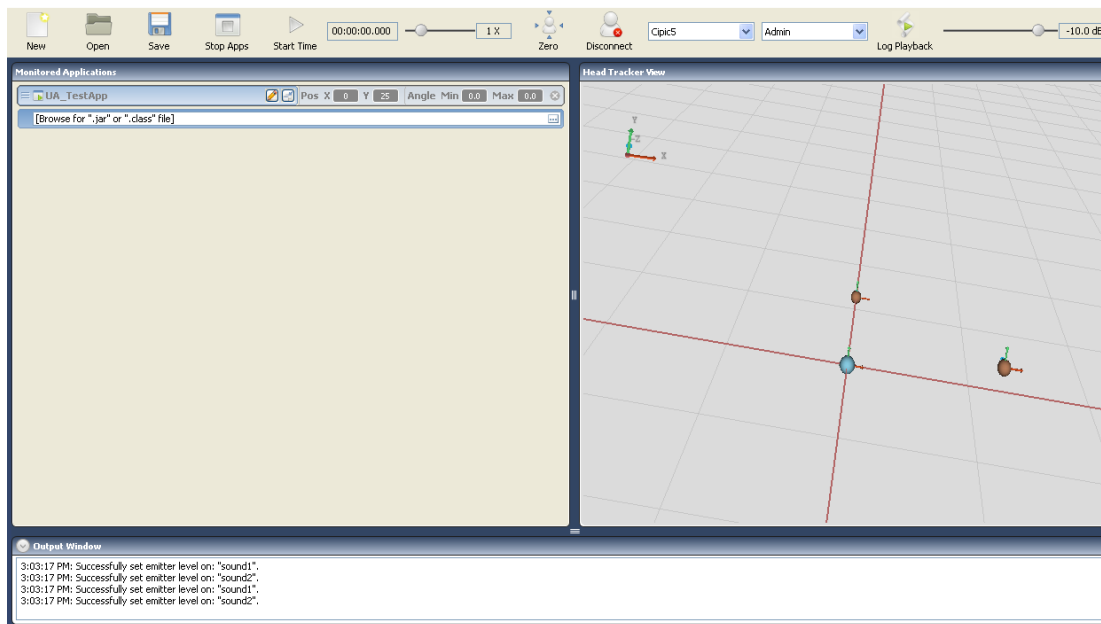


Figure 3. Screenshot of Uber App where the blue object is the observer and the brown objects are the emitters

Executing the project as a whole requires the desktop which is running the main Java program to be in the user's control with the headphones plugged into the desktop. The program also has a User Interface with a mirror image of what is being displayed by the screen. In this mirror image,

the user can either change what is being displayed or zoom in on certain areas in order to focus on a specific area and reduce the number of sounds being played. The laptop will then communicate with the server, constantly adjusting for the different bodies being introduced and removed and the changing orientation of the user's head.

As with any large data set, there are many data points that stand out from the rest of the data points.

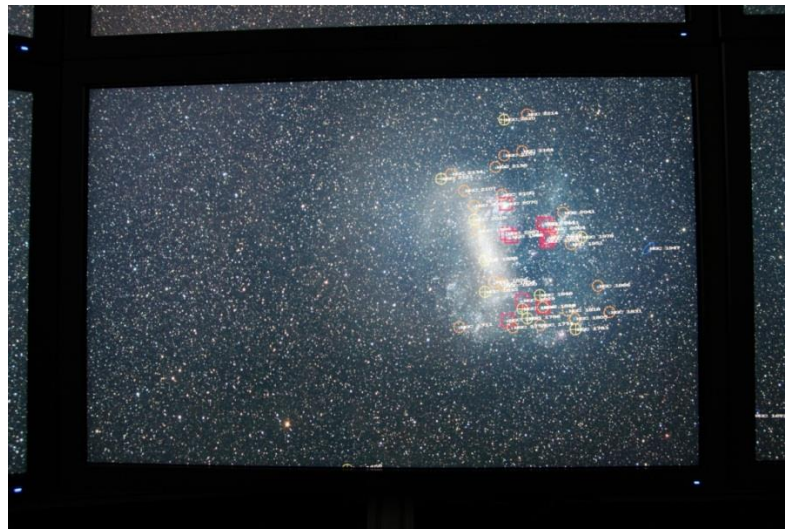


Figure 4. An example of a portion of the panorama display with multiple bodies in the same vicinity

Since in this case, these data points were represented by a sound being played from the apparent location of the data point on the panorama display, there is almost always more than one sound being played at the same time. This leads to the possibility that there will be too many sounds being played for the user to react and process the data presented in the most efficient manner. Thus, it was important to recognize that there is a certain limit as to how many sounds can be playing at the same time before the user's reaction time is less efficient than desired.

Another important part of this project was the type sound that would be played for the different types of data. Since each of the astronomical bodies had common characteristics, I divided these categories into different sublevels where each body would fall into a sublevel. Not

only did I have to figure out a proper sound to play for each category, but there had to be a different sound for each category. A study done by Richard Kronland-Martinet found specific properties of the sounds of three common materials, wood, metal, and glass.¹ With these properties, Kronland-Martinet synthesized various versions of each material, while also tuning each version to the same chroma, which is the quality of a pitch. These varying versions of three sounds all tuned to the same chroma could have been enough for this project, but Kronland-Martinet also took these tuned sounds and fused them together to give an effect of a sound transitioning between two of the three materials.

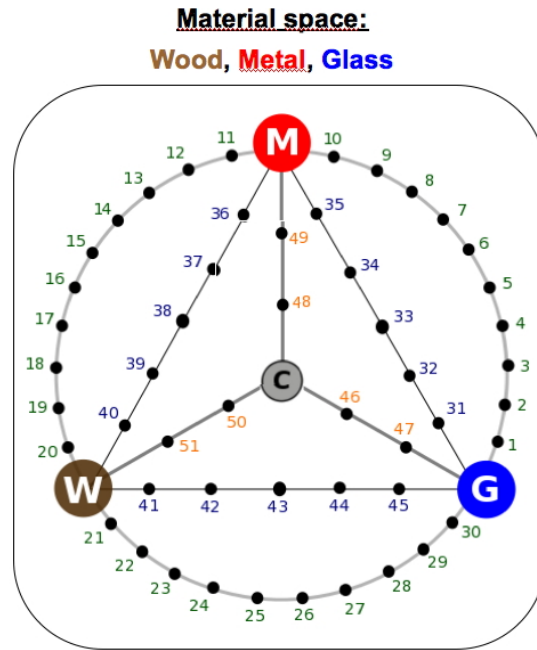


Figure 5. Diagram that shows the way that Kronland-Martinet fused the three distinct sounds

This was especially suiting for this project due to the fact that each body could have more than one characteristic that distinguishes it from the other surrounding bodies, so two sounds would have to originate from the same location. Thus, instead of having two sounds originating from the same location, there could instead be one fused sound originating from the same location which also increases the efficiency of the user recognizing that there is more than one

characteristic in that location. However, at the same time, as described by Hermann, Hunt, and Neuhoff, the sounds being played have to be chosen carefully because the user needs to be notified that there is a body that could be of interest, so the sound needs to be able to get the user's attention, but at the same time, the sound should not annoy the user.² Ultimately, I combined what I learned from both studies in order to find distinct and attention-grabbing sounds and altered them slightly for the different subcategories.

Conclusion

In the end, I did create an effective way to portray the galaxy provided in the Milky Way Galaxy Way set with the use of 3D audio. If I were to continue this research, I would first make this project more portable, such as making it able to run on a tablet. Another possible improvement could be implemented in the tracking system. Currently, the InertiaCube3 only detects head rotation, but does not detect any change in the actual location of the head, which would require the user to stand in the same position instead of being able to move around.

Acknowledgements

I would like to thank my mentor, Dr. Mark Livingston, for giving me the opportunity to work on this project, Mr. Derek Brock and Mr. Brian McClimens for assisting me with the various components of the project related to communicating with SoundScape3D, Mr. Jonathan Decker for helping me with establishing a method of communication between the cluster and the rest of my program, Dr. Zhuming Ai for helping me with the tracking system, the SEAP coordinators for giving me this wonderful opportunity to work at NRL, and my family for helping me throughout the entire process.

Bibliography

¹ Kronland-Martinet, Richard. "Sounds." Sounds. Laboratoire De Mécanique Et D'Acoustique, n.d. Web.

² Hermann, Thomas, Andy Hunt, and John G. Neuhoff. The Sonification Handbook. Berlin: Logos Verlag, 2011. Print.