Astronomy Sound of the Month
Hubble Ultra Deep Field Sonification

Greg Salvesen — gregsalvesen@gmail.com

The *Hubble Space Telescope* stared at one dark patch of sky for over 11 days. The result is the Hubble Ultra-Deep Field (HUDF): an image of 10,000 galaxies! A few years ago, a research team of astronomers measured the distances to all of these galaxies as far away as 13.1 billion light-years. We are seeing these most distant galaxies as they were when the Universe was only about 800 million years old, which is analogous to obtaining a snapshot of a 50 year old person when they were three years old.

This sonification is inspired by my experience at public open house observing nights, where the most common question I get asked is, "How far away is that?" Usually, I just take a sensible guess at a distance and say it confidently. Sure, the HUDF is a super pretty image of 10,000 galaxies, but don't you want to know how far away those galaxies are from us? The goal of this sonification is to interactively move your mouse over any of the HUDF galaxies and have a sound play that indicates its distance.

1 Data

I downloaded the high-quality 2300×2100 pixel JPEG 6.9 MB press release image of the HUDF (`http://hubblesite.org/image/3380/news_release/2014-27`). This is a pretty picture, but it contains no information about the distances to the individual galaxies. For that, I went to the Hubble UVUDF Catalogs website (`https://asd.gsfc.nasa.gov/UVUDF/catalogs.html`) and downloaded both the binary FITS data file `uvudf_rafelski_2015.fits` and the segmentation map FITS file `segmentation_map_rafelski_2015.fits` (Rafelski et al. 2015; `https://arxiv.org/abs/1505.01160`).

The relevant pieces of information contained in the data file for each galaxy in the HUDF are: a unique galaxy ID number and up to three redshift measurements (two photometric, one spectroscopic). The segmentation map is a 2D image whose pixel values correspond to galaxy ID numbers. Pixels in the segmentation map that are not associated with a galaxy have zero value.

2 Visualization

A straightforward process for going from hovering your mouse over a galaxy in the HUDF image to playing the note that represents the distance to that galaxy would go something like this:

1. Move the mouse over a galaxy in the HUDF image and retrieve the (x,y) pixel coordinates.
2. Given these coordinates, lookup that galaxy's ID number from the segmentation map.
3. Given the galaxy's ID number, lookup its redshift from the data file.
4. Convert this redshift to a distance.
5. Map this distance to a pitch and play that note.

My solution wasn't quite this straightforward, but the spirit is the same and I'll describe it briefly.

The crux of the problem is going from Step 1 to Step 2. The galaxies in the HUDF image need to be mapped to their corresponding galaxy ID numbers in the segmentation map. Why is this important? Well, the (x,y) pixel values in the HUDF image are just some pretty combination of (R,G,B) values that the user sees. But the redshift information is linked to the galaxy ID numbers, which are the (x',y') pixel values in the segmentation map. The problem is that $(x,y) \neq (x',y')$ because the HUDF image and the segmentation map have neither the same native resolution nor alignment. In order to get the galaxy ID number for a galaxy in the HUDF image, we need to map (x,y) in the HUDF image to (x',y') in the segmentation map.

One of the first things I did was a pixel-by-pixel transformation between the HUDF image and the segmentation map. In other words, if I have coordinate (x,y) in the HUDF image, what (x',y') coordinate does that correspond to in the segmentation map? In an `iPython` session, I opened up the HUDF image and the segmentation map side-by-side. By eye, I found the (x,y) coordinates for three small galaxies in the HUDF image and the corresponding (x',y') coordinates in the segmentation map. I then solved the six-parameter coordinate transformation matrix with the `Mathematica` notebook `solve_2D_coord_transform.nb`, which is unfortunately not free software. Knowing how to map an (x,y) pixel in the HUDF image to an (x',y') pixel in the segmentation map, I can now generate a segmentation map with the same native resolution and alignment as the HUDF image. This allows me to go from Step 1 to Step 2.

Remember, the segmentation map is a 2D image with each pixel value mapping to either a galaxy ID number or "empty" space. The data file gives the redshift corresponding to each galaxy ID number. Therefore, it is straightforward to turn a 2D image of galaxy ID numbers into a 2D image of redshifts (Step 3). Using the procedure described in the Sonification section below, I convert redshift to distance (Step 4) and then to pitch in order to produce a 2D image of notes. This file is called `NoteMap_large.csv` and is produced with the `Python` script `rescale_segmap.py`. Given an (x,y) pixel coordinate from the HUDF image (Step 1), the contents of the file `NoteMap_large.csv` determine which note to play (Step 5).

Figuring out how to actually implement this full process on the AstroSoM website was really challenging for me. It was a lot of trial and error and I had to learn some `HTML`, `CSS`, `PHP`, and `Javascript`. For fun, I also implemented a magnifying glass that makes it easier to move your mouse over a specific galaxy and explore the HUDF. Clicking on the HUDF image on the AstroSoM homepage takes the user to the sonification for the full resolution HUDF image.

3 Sonification

The data file provides the redshift z for every galaxy. When a spectroscopic redshift is available, I use that. When two photometric redshifts are available, I use their average. To convert a galaxy's redshift into a distance, I first calculate the corresponding "lookback time" $t_{\rm lb}(z)$ with the equation (Hogg 2000; https://arxiv.org/abs/astro-ph/9905116),

$$t_{\rm lb}(z) = \frac{1}{H_0} \int_0^z \frac{dz}{(1+z)\left[\Omega(1+z)^3 + \Omega_k (1+z)^2 + \Omega_\Lambda\right]^{1/2}}, \tag{1}$$

where I adopted the cosmological parameters: $H_0 = 69.6$ km s^{-1} Mpc^{-1}, $\Omega = 0.286$, $\Omega_k = 0$, $\Omega_\Lambda = 0.714$ (Bennett et al. 2014; https://arxiv.org/abs/1406.1718). To get the distance to the galaxy, I multiply the lookback time by the speed of light, $D = t_{\rm lb} \times c$. I should stress that these distances are just sensible guesses because nearly all of them are determined from a photometric redshift, as opposed to a much more reliable spectroscopic redshift.

After calculating the distance to every galaxy, I generated all 128 MIDI notes using `MIDITime` (https://github.com/cirlabs/miditime) and the script `midi_notes.py`. Each note has a 0.5 second duration and is output as an MP3 and OGG file. Distances to HUDF objects (galaxies and a few stars) range from zero to 13.1 billion light-years. So, I started from the lowest note on a piano A_1 (MIDI #21) as representing zero distance and mapped each chromatic half-step up in pitch to a distance increment of 150 million light-years until I reached $D = 13.05$ light-years, or the note C_9 (MIDI #108). For a galaxy at some distance, the note closest to that distance is sounded.

4 Lessons Learned in the Process

- Very low MIDI notes suffered from aliasing, so it made sense to just adopt the lowest note on the piano as my zero distance reference.
- Saving note filenames containing the "#" character ("#" = "sharp") is a bad idea.
- I used `MIDITime` because I am familiar with it, but this was overkill for simply generating a bunch of individual MIDI notes.
- The high-resolution image is very large (2300×2100 pixels) and the medium-resolution image displayed on the AstroSoM homepage is a good size, too (800×730 pixels). Loading in this 2D list of pixels was frustratingly slow, but using `PapaParse` to load in the `NoteMap_large.csv` file in `Javascript` seems to work nicely.
- I played around a lot with `HTML5` Canvas, initially hoping to stack the HUDF press release image on top of the segmentation map and have these layers communicate with each other. This didn't work because I didn't understand that when you tell Canvas to return the pixel (R,G,B,A) values, say, it does that for whatever is displayed to the screen. To my knowledge, Canvas cannot peel back a layer from what is displayed and retrieve that information.
- While developing this HUDF sonification, I was creating the AstroSoM website simultaneously. During this process, my original vision for AstroSoM became more ambitious because I became aware of some of the neat interactive stuff that can be done with web development tools. Hopefully I'll continue to learn and make AstroSoM a fun little website that promotes astronomy research to an audience of normal people.
- I managed to eventually solve all of my `HTML`/`CSS`/`PHP`/`Javascript` problems with diligent Googleing.
- Changing all PNG files to JPG files with 50% "image quality" on http://resizeimage.net/ had a huge impact on reducing the file size (allowing for faster load times) without noticeably dinging the image quality.