Visualizing sound on the surface of a sphere.

Marco M. Grond

1 Introduction

Analysis of musical compositions, especially those composed before the current era of contemporary music, have traditionally taken place by analysing and interpreting the score of a composition. However, the current movement in contemporary music is to take all aspects of a musical performance into account, including improvisations, physical properties of the instruments used during the performance and ambient noises that occur within the concert area. Because of this, much information about the piece is not captured in the score, but rather has to be experienced. This causes many difficulties when attempting to analyse a piece of music, and makes it almost impossible for persons who have not experienced a performance to analyse it.

Analysing musical compositions allows for a better understanding of the piece itself. It is often hard to capture exactly what a piece is meant to convey without being physically present for a performance, but through sound recording a lot of information concerning the production can be captured. However, analysing this data can be quite tedious and time consuming, especially if multiple simultaneous recordings are made in an attempt to capture more aspects of the performance.

Ambisonics, an area pertaining to the study of sound around a full sphere, opens up the door to capture the aspects of a musical performance that are not present on the score or needs to be experienced to fully capture it. This is a fairly new research are, but research into this area could lead to not only a better platform for the analysis of spatial music, but also lead to improvements to popular immersive technologies such as audio reproduction in cinemas or virtual reality.

In this project I attempted to visualize the data captured by a microphone array spread across the surface of a sphere. This required the interpolation of data across the sphere, since only a small number of microphones are present and the data was quite sparse. Different interpolation techniques were employed to accomplish this task, and afterwards compared to one another in an attempt to understand which would be best suited for this application. Different visualization techniques were additionally applied to this interpolated data, to see which would best convey the trends in the data as well as give an overall idea of what was happening at different times throughout the musical performance.

A motivation of the problem and its importance as well as some background information on sound analysis and previous attempts to interpolate data on a sphere follow this section. Next, a detailed discussion of the implementation of the different techniques are presented, followed by the results obtained by these techniques. Finally, the planned future work as well as a short conclusion are given.

2 Motivation

The recording and reproduction of sound in three dimensions is an emerging area within the broader field of sound processing and analysis, which has seen increased interest over the last couple of years due to its relevance to enhancing experiences in immersive technologies, such as audio reproduction in cinema and virtual reality. Further analysis of this data, such as visualizing how sound behaves in three dimensional space could lead to the betterment of techniques and research that would benefit these technologies. Furthermore, this research could lead to advances in the location in three dimensional space of sound sources, in cases where visual data might not be present.

Visualizing sound in three dimensions is an interesting but difficult problem, due to the dimensionality of the ambisonic dataset, which consists of the two dimensional location where the sound recording was made, the time dimension of the sound recording and the changes in air pressure caused by the sound over time. This data can be further expanded by computing power spectra at different time steps, further increasing the dimensionality and the difficulty in visualizing this data. As of yet, no ideal solution has been found on how to best visualize this data, leaving a lot of room for research into better visualization techniques.

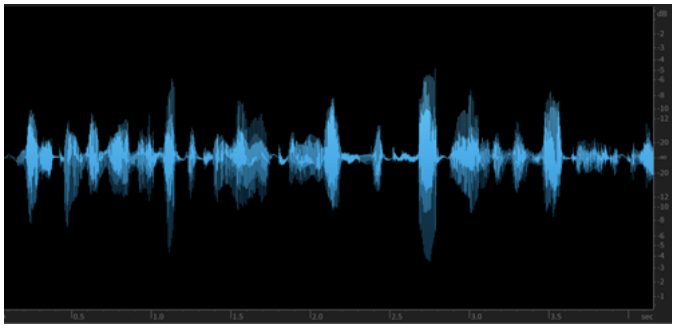


Fig. 1: An example of the waveform with time along the x-axis and differences in air pressure caused by the sound waves along the y-axis [1].

Using a microphone array to record sound adds an extra spatial dimension to the recorded audio, which in turn results in additional data for the analysis of music. Visualizing this data can be quite difficult however, since every microphone records a separate audio signal, and visualizing how these signal combine on the surface of a sphere or interact with one another can be somewhat challenging. From a musicology standpoint, the results of these visualizations could help improve the analytical process surrounding a musical piece, as well as help composers or performers to set up the performance area to best convey their music to the audience. Visualizing this data could also help better understand what the audience is actually experiencing, as opposed to the expectations.

3 Background and related work

Traditionally, sound has been visualized in two dimensions, with the most well known sound visualization technique being the waveform, an example of which is shown in Figure 1. This visualization conveys little information about a sound piece, other than the change in air pressure caused by the sound waves over time. The next type of visualization, which is more prevalent when doing sound analysis and research on sound, is the spectrogram which was first used to visualize sound by Koenig *et al.* [7], an example of which is presented in Figure 2. This type of visualization presents much more information about a piece of sound than the waveform, since it showcases the changes in the power spectra of different frequencies over time, which has lead to it being the standard way of visualizing sound pieces for analysis and research. *TIAALS*, a tool which facilitates the analysis of electroacoustic music, is a prime example of how spectrograms can be used in the research and analysis of sound, in particular musical pieces [4]. Other visualization techniques that have been employed to analyse musical pieces, are self similarity and beat spectra, which attempt to find similar structures at different time intervals within a musical piece using some distance function, and visualizing the results as two dimensional plots [5].

The em32 Eigenmike [9], which was used to acquire the data used in this project, is a microphone array which consists of 32 microphones on the surface of a sphere with a radius of 42mm. Each microphone

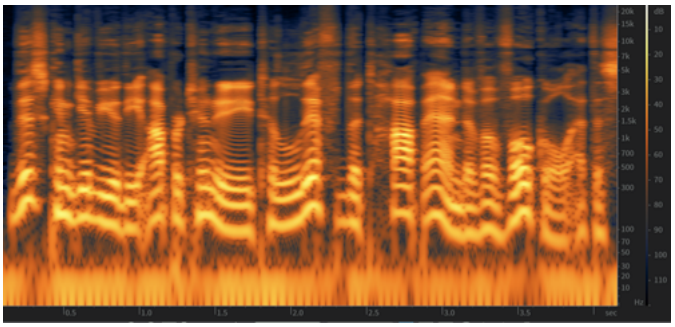


Fig. 2: An example of a spectrogram generated from the waveform in Figure 1, with time along the x-axis and frequency along the y-axis [1].

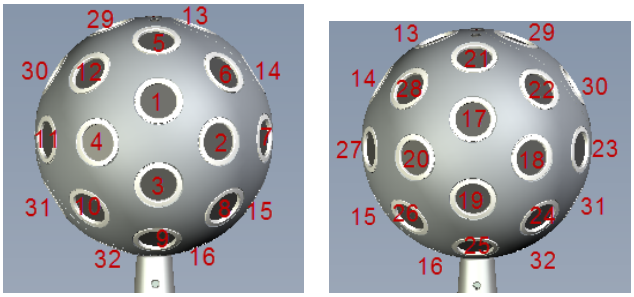


Fig. 3: A view of two sides of the Eigenmike.

records audio data at a rate of 44.1kHz as a separate channel. The 32 microphones can be connected to form a grid consisting of 60 triangles. These triangles are not congruent however, due to some physical features of the Eigenmike, and do not form a regular platonic solid. Two views of the Eigenmike are shown in Figure 3, while the solid formed by connecting the locations of the microphones is shown in Figure 4. The actual locations of the microphones are given in Table 1.

Ambisonics is a relatively young field and as such not much research has gone into the visualization of this type of data. Interpolation of data on the surface of a sphere however, has seen more research and multiple techniques have been developed to accomplish this goal. The techniques can be grouped into three different categories, namely linear interpolation, distance based interpolation and cubic interpolation [3].

Linear interpolation is the simplest and least expensive interpolation method that is used to interpolate data on a sphere. Linear interpolation

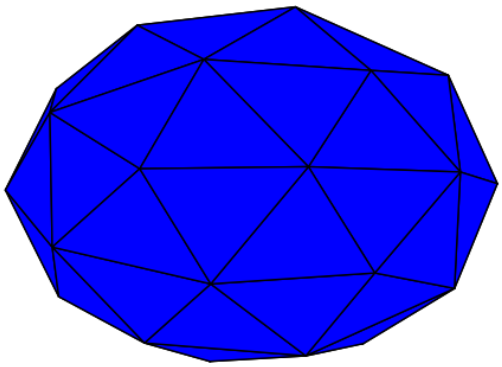


Fig. 4: The solid formed by connecting the microphone locations of the Eigenmike to form a triangular grid.

Table 1: Azimuth (ϕ) and elevation (θ) angles, given in degrees, for all microphones on the surface of the Eigenmike.

Microphone #	θ	ϕ	Microphone #	θ	ϕ
1	69	0	17	69	180
2	90	32	18	90	212
3	111	0	19	111	180
4	90	328	20	90	148
5	32	0	21	32	180
6	55	45	22	55	225
7	90	69	23	90	249
8	125	45	24	125	225
9	148	0	25	148	180
10	125	315	26	125	135
11	90	291	27	90	111
12	55	315	28	55	135
13	21	91	29	21	269
14	58	90	30	58	270
15	121	90	31	122	270
16	159	89	32	159	271

divides the sphere into a grid of cells and interpolates a value within the cell based solely on the sample points on the corners of the cell. Most of the linear interpolation techniques on a sphere implement a barycentric coordinate system where sample points are connected to form a grid of triangles. The barycentric coordinate system is then used to determine the weights as a combination of the areas of the triangles formed by connecting the interpolation point to the original sample points. Different techniques either map the spherical triangle to a planar triangle to interpolate the points, or use the linear barycentric coordinates when interpolating [3, 8].

The distance based scattered data interpolation techniques use all of the sampled points when computing the value at any point on the sphere. This allows for a smoother final function over the entire sphere. The biggest challenge with the distance based interpolation techniques is to compute the weights for each of the sampled values, which take the distance between the point of interest and each sample point into account. This allows points closer to the interpolation location to have a greater influence on the final value than points further away. The two most common distance based scattered data interpolation techniques that are applied on spheres are Shepard's method [12], which computes the ratio of the distance between the interpolation point and a sample point and the sum of the distance to all sample points as the weights, and radial basis functions [2] which choose a positive definite function as the interpolation basis function and computes the weights by solving a linear system to ensure that the interpolated value at the sample points produce the correct values.

The final interpolation type, cubic interpolation, once again requires that the sphere be subdivided into a grid of cells. Rather than using only the sample points of the cell within which the interpolation point falls, the sample values of cells around this cell are also used to compute the interpolated value [3]. Cubic interpolation can become quite expensive to compute on these spherical grids, and as such was not explored in this project.

4 Implementation Details and Methodology

My implementation focussed on the data processing and data mapping parts of the visualization process. The data used for this project was made available by my supervisor, Dr. Jeffrey Boyd, and had already been collected before the start of this project. The visualization of the final results was done using Mayavi [11], a Python wrapper for VTK.

4.1 Data preprocessing

The raw data that was used during the course of this project consisted of a 10 second long audio file recording of a performance of *Touch* by Keith Hamel [6], recorded using a em32 Eigenmike microphone array. The sound file was saved in a 'wav' format which consisted of the changes in air pressure, sampled at a rate of 44.1kHz. The file

also contained 32 distinct audio channels, each recorded by a different microphone on the surface of the Eigenmike.

The raw data is not ideally suited for a visualization, since the changes in air pressure and the high sampling rate would cause flickering when attempting to visualize this data as a video or time series. To remove this occurrence and produce smoother visualizations, the data was downsampled through averaging the magnitude of the changes in air pressure and smoothing the downsampled data through cubic interpolation. This process was applied to each sound channel individually to end up with 32 smoothed signals which captured the shape of the data and were used for the subsequent parts of the project. An example of this process on a single channel is shown in Figure 5.

4.2 Interpolation techniques

To find the values of points on the sphere, three different scattered data interpolation techniques on the surface of a sphere were employed. These techniques are linear interpolation, Shepard's interpolation and interpolation using radial basis functions.

4.2.1 Linear interpolation

The first interpolation technique that is used to determine the value of a point on the sphere is linear interpolation. To implement linear interpolation, the sphere has to be subdivided into multiple different cells, with the sample points on the corners of the cells. The interpolated value at a point within the cell is then calculated as the weighted sum of the sample values on the cell's corners. Since the sample points on the sphere are the microphone locations, the sphere was simply divided into multiple triangles with their corners at the microphone locations. Altogether there are 32 different microphones with 60 triangles formed by connecting them together. Rather than using the spherical triangles formed by connecting the microphone locations with arcs, the microphones were instead connected by straight lines which formed triangles with flat surfaces. These triangles served as the cells in which linear interpolation took place.

Simple barycentric interpolation was employed to determine the value at a point within the cell. Given a triangle with corners p_1, p_2 and p_3 with sampled values f_1, f_2 and f_3, the value at a point p within the triangle is given by:

$$r = \frac{\text{Area of } \Delta p_1 p p_3}{\text{Area of } \Delta p_1 p_2 p_3},$$
$$s = \frac{\text{Area of } \Delta p_1 p_2 p}{\text{Area of } \Delta p_1 p_2 p_3},$$
$$\tilde{f}(p) = (1-r-s)f_1 + rf_2 + sf3,$$

where $\tilde{f}(p)$ is the interpolated value at the point p. The triangles are recursively subdivided into 4 smaller triangles to find multiple points spread across the surface of the triangle on which to interpolate. Once the value has been determined at every point within a triangle, these values are mapped onto the sphere by drawing a ray from the origin of the coordinate system through the interpolated point and assigning the interpolated value to the intersection point between the ray and the surface of the sphere.

4.2.2 Shepard's interpolation

The second interpolation technique that was implemented, is a distance based scattered data interpolation technique known as Shepard's interpolation. Whereas linear interpolation only takes the values at the corners of the cell into account when interpolating at a point within the cell, Shepard's interpolation takes all sample points on the surface of the sphere into account when interpolating on any point on the sphere. This is done by computing the interpolated point as the weighted sum of all of the sampled values. The weights used in this interpolation technique depend on the geodesic distance between the point of interest and the corresponding sample point. The weights are computed in such a way that the sample point closer to the point of interest have a higher weight than those further away. The formula for interpolating the value at a point p is given by:

$$\tilde{f}(p) = \sum_k^N \frac{w_k(p)}{\sum_j w_j(p)} f(p_k),$$
$$w_i(p) = ||p - p_i||^{-c},$$

where $||p-p_i||$ is the geodesic distance between the point p and the sample point p_i, $f(p_k)$ is the actual value at a sample point p_k and c is a positive real number. For my implementation c was assigned a value of 2. The fact that c is a positive real number led to some complications during the implementation of the algorithm, since the weight between a point and itself would result in division by zero. To get around this fact, the values at the sample points were simply set to the actual sampled value. The geodesic distance between two points were computed as:

$$||p - p_i|| = \text{atan}\left(\frac{|n \times n_i|}{n \cdot n_i}\right),$$

where n and n_i are the normals to the sphere at points p and p_i respectively. The points used in Shepard's interpolation are also on the surface of the sphere rather than on the flat triangles connecting the sample points, which means that they do not have to be mapped onto the sphere, as was the case for linear interpolation.

4.2.3 Interpolation using radial basis functions

The final type of interpolation that was implemented, interpolation using radial basis functions, is another example of a distance based scattered data interpolation technique. This means that, similar to Shepard's interpolation, the value at any point is computed as the weighted sum between basis functions for all of the different sample points on the surface of the sphere. The geodesic distance is once again used in this interpolation technique, but this time the distance is used as the variable for the basis function, rather than for the weights. There is no set rule on what constitutes a good basis function, and for this implementation a Gaussian basis function is used. Once again the points are sampled on the surface of the sphere rather than mapping them onto the sphere from some flat surface. The interpolation function as well as the Gaussian basis function are given by:

$$\tilde{f}(p) = \sum_k^N w_k(p)\phi(||p - p_k||),$$
$$\phi(s) = e^{-\left(\frac{s}{c}\right)^2}.$$

where $\phi(s)$ is the basis function and w_k are the weights for the sample values, p_k. The constant c in the basis function was assigned a value of 1. The weights are computed by solving the linear system:

$$\forall p_i \rightarrow \tilde{f}(p_i) = \sum_k^N w_k(p_i)\phi(||p_i - p_k||) = f_i,$$

where f_i are the actual sample values for points p_i. One limiting factor on the basis function is that it should be positive definite in order to ensure that the matrix used to solve the linear system is non-singular. Although solving this linear system requires some extra computations, it only has to be computed once for every time step rather than for every interpolation point.

4.3 Visualization techniques

To visualize the data on the surface of the sphere, multiple techniques were employed. The first technique used a colourmap to visualize the different scalar values on the surface of the sphere, the second technique double encoded the scalar values with both colour and distance from the centre of the sphere and the third technique double encoded the values with colour and a density plot on the surface of the sphere.

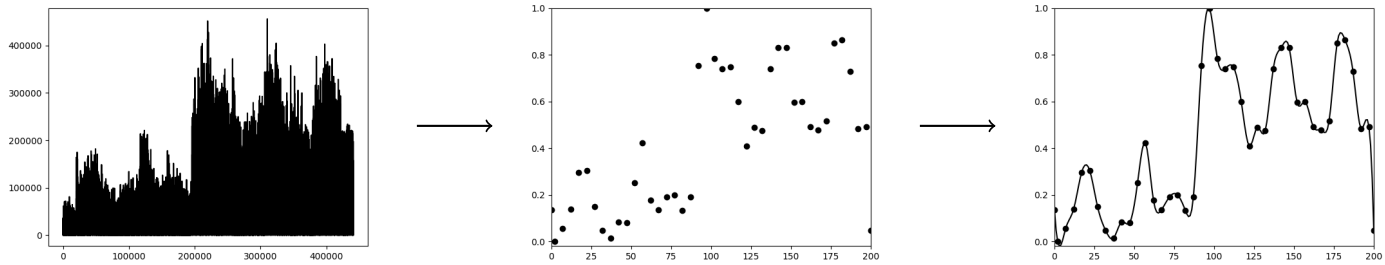


Fig. 5: Visualization of the downsampling and smoothing of a single audio channel. The figure on the left shows the magnitudes of the changes in air pressure, the central figure shows the points obtained from downsampling these magnitudes and the figure on the right shows the smoothed function obtained by applying cubic interpolation on the sample points.

Fig. 6: The cool-warm colourmap used in the visualizations of the scalar data.

4.3.1 Colourmapping

The first visualization technique simply applied a colourmap to the values on the surface of the sphere. During the data processing steps of this project, all interpolated values were scaled to be between 0 and 1, which simplified the implementation of the colourmap. A cool-warm colourmap [10], shown in Figure 6, was used, with blue corresponding to low values and red corresponding to high values. Altogether, between 4000 and 5000 points on the sphere were interpolated using the interpolation techniques discussed in the previous sections, after which the interpolated results at these points were visualized by mapping them to the colourmap.

4.3.2 Colourmapping with height

The second visualization technique employed a double encoding of the scalar values by mapping each value to a colour as well as the distance between the point on the sphere and the centre of the sphere. Once again, between 4000 and 5000 points are interpolated on the surface of the sphere and mapped to the cool-warm colourmap. This time however, the interpolated value is additionally mapped to the distance between the centre of the sphere and the point, by setting the distance to be equal to:

$$r(p) = c + \tilde{f}(x)^2,$$

where c is a constant for the minimum distance from the centre of the sphere, and was implemented to be 5 units. Essentially, this means that the visualization stretches the sphere out in areas that have high values, while keeping areas with low values closer to the sphere's original radius.

4.3.3 Colourmapping with density plots

The final visualization was once again a double encoding of the scalar value, this time using a colourmap along with a density plot. The density plot places small glyphs on the surface of the sphere, where the amount of glyphs in an area is dependent on the interpolated value in that area. Additionally, each glyph is coloured according to the cool-warm colourmap.

To apply the glyphs, the surface of the sphere is approximated by 960 triangles by recursively dividing the spherical triangles formed by the locations of the microphones on the sphere into 4 smaller triangles. The interpolated value at the centre of each of these triangles is then computed and divided into 10 bins. The number of glyphs placed within a triangle is directly related to the bin that its interpolated value was divided into, where a lower interpolated value would result in less glyphs being placed within the triangle. The correct number of glyphs are then randomly placed within the approximating triangle.

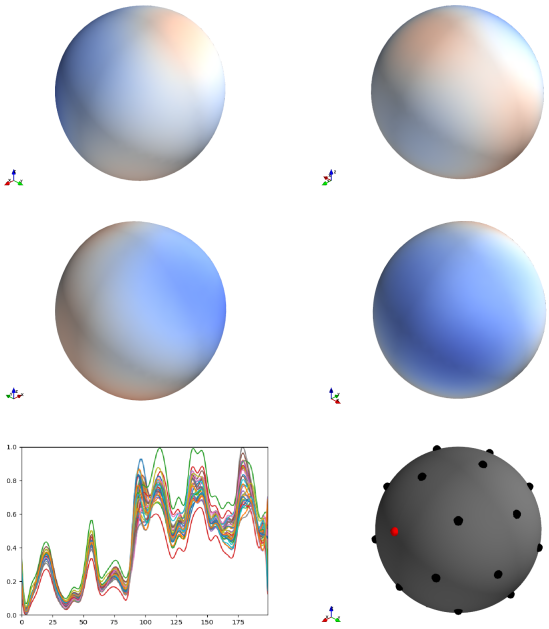


Fig. 7: A still frame from one of the video visualizations. The top four images are different angled views of the same sphere visualization, while the graph in the bottom left shows the smoothed waveform for each of the different audio channels. The image on the bottom right simply shows the microphone that recorded the sound which is played in the video.

4.4 Videos

Videos were also computed for each of the interpolation and visualization combinations. The videos were created by computing the visualizations at different time steps and combining the resulting images. The videos are available at `https://www.youtube.com/playlist?list=PL-5kLq3t5DHx6u151c2lGbQ1H-ZkOe-Ho`. These videos were generated by sampling the original data twice per second and computing 40 frames per second for the visualization. A still frame of one of the videos is shown in Figure 7.

5 Results and Discussion

All of the different visualization techniques were applied to the results of each of the interpolation techniques. This section will present,

discuss and compare these different visualization and interpolation techniques to one another.

Visualization using only colourmaps for linear, Shepard's and radial basis function interpolation are shown in Figure 8. These figures clearly show one of the biggest differences in quality between the different interpolation techniques. The visualization of the data computed through linear interpolation shows some lines between the sampling points, especially when there is a relatively large difference between the actual values at the sample points on the corners of a cell. Shepard's interpolation also leads to some artefacts, but in this case they are only visible at the sample point locations. This is possibly due to the fact that this interpolation technique takes all of the sample points around the sphere into account. The visualization on the data generated through radial basis functions does not seem to exhibit these types of artefacts, and overall seems to result in a smoother transition between colours on the surface of the sphere.

Visualization using a combination of a colourmap and distance from the centre of the sphere was the second type of visualization attempted. The results of this visualization applied to all interpolation methods are shown in Figure 9. Once again, some of the differences between the different interpolation methods are made clear in these examples. The linear interpolation led to a more rounded final result, most likely due to the fact that only a small percentage of the total number of sample points were used in the interpolation of any given point. Shepard's interpolation led to a more lumpy visualization, likely caused by the same artefacts visible in the colourmapping visualization, where the values at the sample points were noticeably different from the interpolated values. The radial basis functions once again produced the smoothest visualization, and accentuated the areas with higher values by stretching the entire sphere in the correct directions.

The final visualization using a density plot along with a colourmap is shown in Figure 10. The differences between the different interpolation techniques are harder to recognize in this visualization technique. The figures do show however, that the radial basis function interpolation tends to stretch higher values over a greater are than the other two interpolation techniques. The density plot visualization is somewhat misleading for the two dimensional images, since the curvature of the sphere leads to the appearance that glyphs closer to the side of the sphere are more densely packed together when compared to the middle of the sphere, even if the interpolated values are more or less the same. The density plots also seem to be more suited to static images, since the random element leads to some flickering when the images are combined into a video.

A comparison between all of the different visualization and interpolation combinations is presented in Figure 11. This figure suggests that the double encoding of the scalar values tend to improve the overall visualization and leads to a more intuitive understanding of the data. The visualization which seems to be most visually appealing and clearly communicates the trends in the data is the double encoding using colourmaps and distance from the centre of the sphere, likely due to the intuitive nature of this visualization.

Important to note is that the artefacts which appear in the different visualizations of the interpolated data are not noticeable when the visualizations are combined into a video, since these artefacts only appear in certain frames within the video and aren't frequent enough to truly be noticeable. Another observation is that the videos seem smoother when the waveforms are downsampled to fewer points. Even when 4 or 5 points per second are sampled, the videos tend to exhibit a bit of flickering due to the rapid changes in the smoothed graph and the large gradients that occur as a result of this.

The computation times to generate a single image for the different visualization and interpolation combinations are shown in Table 2. This clearly shows that linear interpolation has the shortest computation time, a fact which becomes very noticeable when a large number of images have to be generated. The computation time for Shepard's interpolation as well as interpolation using radial basis functions are comparable for all of the different visualization techniques. The timing results are somewhat expected, since the linear interpolation uses only 3 sample points to compute the value at any point on the sphere, while the two distance based interpolation techniques use all 32 sample points on the sphere. Finally, the density visualizations were much slower than the other visualizations, possibly due to the large number of glyphs that needed to be computed and visualized.

Table 2: Time (seconds) to generate a single image for different combinations of interpolation and visualization methods.

	Colourmap	Colourmap & Height	Density
Linear interp.	2.2679	2.3039	2.7526
Shepard's interp.	4.7279	4.7728	16.3605
RBF interp.	4.9156	4.9309	16.4399

6 Future work and conclusions

Overall, this project presented some viable options to visualize sound waveforms on the surface of a sphere, while also presenting some strengths as well as weaknesses of the different interpolation techniques which were used. As is the case with most data processing and visualization techniques, there seems to be trade-off between the speed and the quality of the interpolation and visualization. The double encoding seems to convey information about the sound data better, as well as being more visually appealing, with the visualization using colour along with the distance from the centre of the circle resulting in the best visualization.

The next step in this project would be to refine the visualization of the density plot to remove the flickering from the videos produced from this technique. One possible way to do this would be to implement a method which moves, splits and merges a predetermined set of glyphs on the surface of the sphere, rather than randomly placing these glyphs on the sphere for each frame of the video. This would allow for a smoother transition between the frames, and perhaps lead to better communication regarding trends within the data.

Another area that will be explored is the visualization of the spectrograms generated from the waveforms on the surface of the sphere. The interpolation techniques which were discussed in previous sections should be sufficient for this type of data, while the visualization techniques will have to be re-examined. This is due to the high number of scalar values which are present at every time step in the spectrogram data, which would have to be visualized without causing too much visual clutter on the surface of the sphere.

References

[1] Understanding the spectogram/waveform display. `https://www.izotope.com/en/learning/rx/understanding-the-spectrogram-waveform-display.html`. Accessed: 2017-10-18.

[2] M. D. Buhmann. Radial basis functions. *Acta numerica*, 9:1–38, 2000.

[3] M. F. Carfora. Interpolation on spherical geodesic grids: a comparative study. *Journal of Computational and Applied Mathematics*, 210(1):99–105, 2007.

[4] M. Clarke, F. Dufeu, and P. Manning. TIAALS: A New Generic Set of Tools for the Interactive Aural Analysis of Electroacoustic Music. In *Proceedings of the Electroacoustic Music Studies Conference*, 2013.

[5] J. Foote and M. Cooper. Visualizing musical structure and rhythm via self-similarity. In *ICMC*, vol. 1, pp. 423–430, 2001.

[6] K. A. Hamel. Touch. SOCAN, 2012.

[7] W. Koenig, H. Dunn, and L. Lacy. The sound spectrograph. *The Journal of the Acoustical Society of America*, 18(1):19–49, 1946.

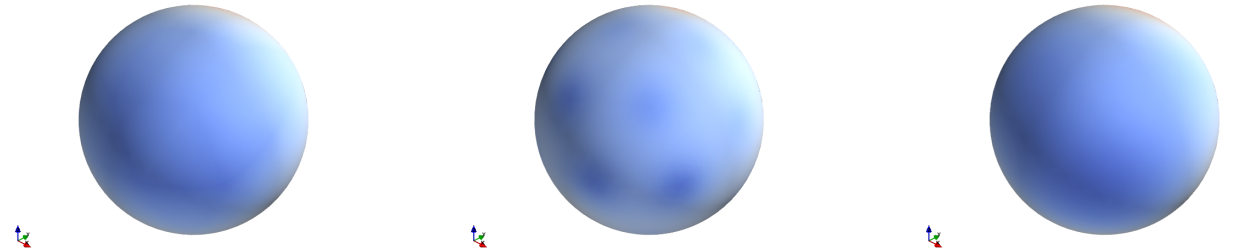
[8] C. L. Lawson. C^1 surface interpolation for scattered data on a sphere. *The Rocky Mountain journal of mathematics*, pp. 177–202, 1984.

[9] mh Acoustics LLC, 25A Summit Ave Summit, NJ 07901 USA. *mh Acoustic User Manual*.

[10] K. Moreland. Diverging color maps for scientific visualization. *Advances in Visual Computing*, pp. 92–103, 2009.

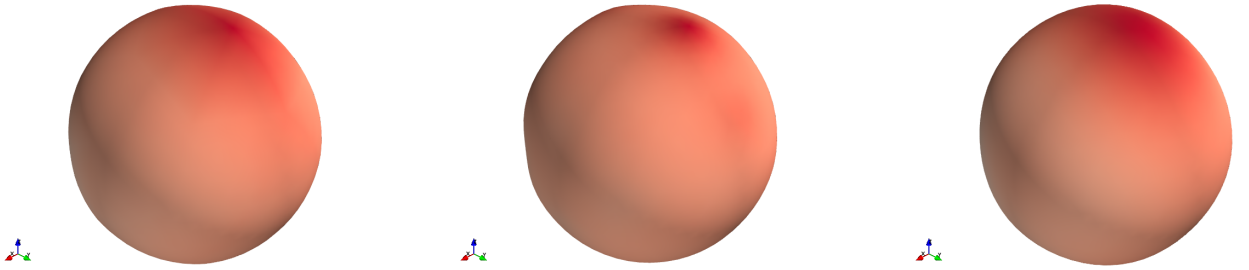
[11] P. Ramachandran and G. Varoquaux. Mayavi: 3d visualization of scientific data. *Computing in Science & Engineering*, 13(2):40–51, 2011.

[12] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524. ACM, 1968.



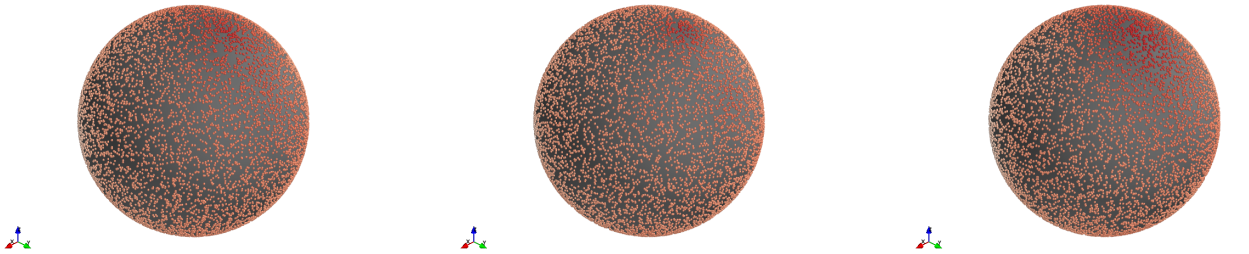
(a) Linear interpolation (b) Shepard's interpolation (c) Radial basis function interpolation

Fig. 8: Visualization using only a colourmap.



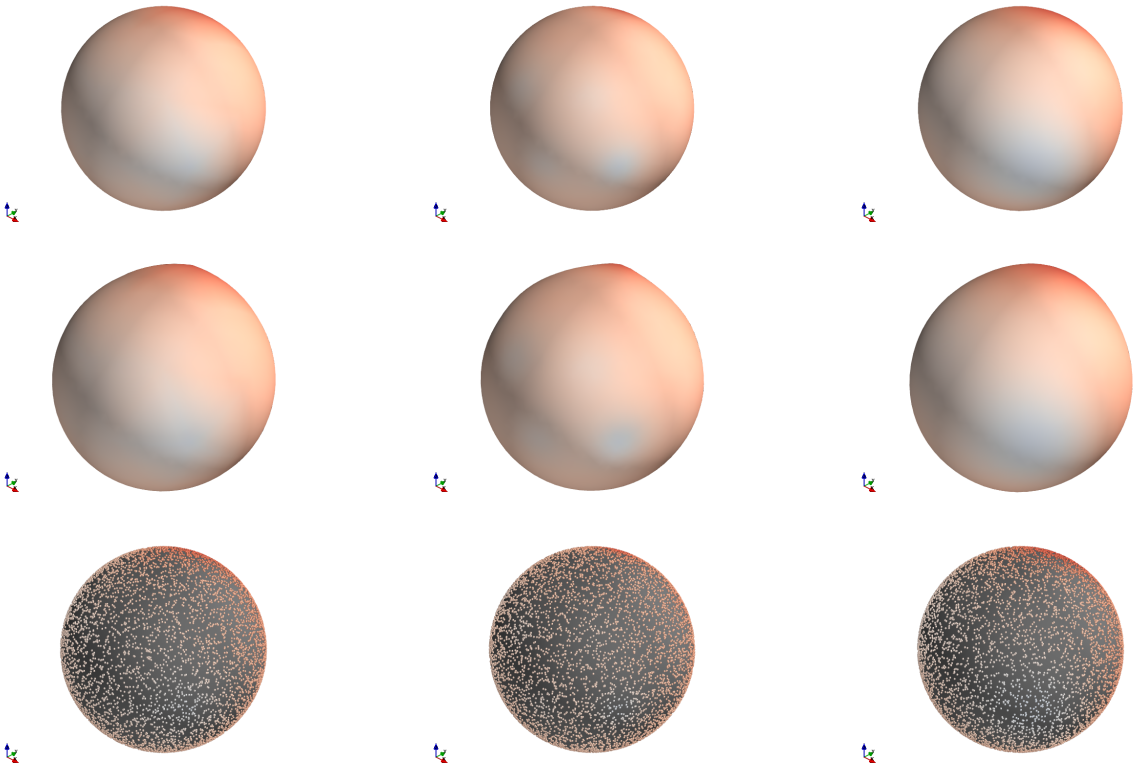
(a) Linear interpolation (b) Shepard's interpolation (c) Radial basis function interpolation

Fig. 9: Visualization using a colourmap as well as distance from the centre of the circle.



(a) Linear interpolation (b) Shepard's interpolation (c) Radial basis function interpolation

Fig. 10: Visualization using a colourmap as well as density.



(a) Linear (b) Shepard (c) RBF

Fig. 11: A comparison between the different visualization and interpolation techniques on the same data. The visualizations in the first row only make use of colourmaps, the second row uses both colourmaps and height and the third row uses colourmaps along with a density plot. The first column used linear interpolation, the middle column used Shepard's interpolation and the third column used radial basis function interpolation.