Designing Mappings for Musical Interfaces Using Preset Interpolation

Martin Marier
Center for Interdisciplinary Research in Music Media and Technology (CIRMMT)
Faculté de musique
Université de Montréal
Montréal (QC), Canada
martin.marier.1@umontreal.ca

ABSTRACT

A new method for interpolating between presets is described. The interpolation algorithm called *Intersecting N-Spheres Interpolation* is simple to compute and its generalization to higher dimensions is straightforward. The current implementation in the SuperCollider environment is presented as a tool that eases the design of many-to-many mappings for musical interfaces. Examples of its uses, including such mappings in conjunction with a musical interface called the sponge, are given and discussed.

Keywords

Mapping, Preset, Interpolation, Sponge, SuperCollider

1. INTRODUCTION

Mapping is a very important and delicate step in designing a digital musical instrument (DMI). This paper presents a tool to design complex (many-to-many) mappings, which are generally considered to yield more interesting results than simpler (*one-to-one*) mappings [7, 11].

This tool, based on preset interpolation, is implemented in the SuperCollider environment as an object named `PresetInterpolator`. It was inspired by the many preset interpolation systems found in other environments. To name a few:

- The GRM tools plug-ins [4];
- Logic Pro's Sculpture synthesizer.
- Max 5's *nodes* object;
- Spain and Polfreman's *Interpolator* [12];
- Momeni and Wessel's *Spacemaster* Max patch [10] and its derivatives [5];
- AudioMulch's Metasurface [1].

A preset interpolation system allows the control of many parameters (of a processing or synthesis algorithm) using a limited number of controllers or in other words dimension reduction. The systems mentioned above allow *one-to-many* or *two-to-many* mappings. This paper explores the possibility of extending these two-dimensional interpolation systems to three or more dimensions which allow *n-to-m* mappings.

The main application for such a mapping strategy would be with interfaces including multiple continuous sensors. If each data stream is considered as a dimension, the interface could be seen as a multidimensional space navigator.

Such a dimensionality reduction is beneficial because it leads to a control that is easier to intuit [6].

NIME'12, May 21 – 23, 2012, University of Michigan, Ann Arbor.

2. NATURAL NEIGHBOUR INTERPOLATION

The Metasurface is a preset interpolation system included in AudioMulch, a software environment for sound synthesis and processing [1]. After data points relating to chosen sound presets are positioned on a two dimensional plane, the user can interpolate between them simply by moving a cursor (the interpolation point) around on the surface. The interpolation method used by the Metasurface is Natural Neighbour Interpolation (NNI). It is explained in detail in [3] and has the following desirable properties.

The interpolated surface is continuous. There are no gaps or sudden steps in the interpolated surface. In gravity models (like Max's *nodes* object), only the data points inside a specified radius are used for the interpolation. If there are no data points inside this region, the interpolation fails and there is a gap in the surface. The light beam model described in [12] also has this problem.

The interpolated surface is continuously differentiable. The surface is smooth; there are no singularities.

The system is autonomous. The only variables are the data points. The user is not required to specify any extra parameters. In the Interpolator [12], the user has to fill three variables for each data point; in the Radial Basis Functions [5], two are necessary. While additional parameters may make a tool more versatile, they also require a better understanding of the underlying algorithm and could distract a user from the primary task.

The data points can be positioned freely. A user can put any number of data points on the surface, and he can position them anywhere.

Interpolation is local. Only a limited number of points (the nearest ones) are used for the interpolation. This is unlike in a simple inverse distance weighting system where all data points are always taken into account, which can slow down the calculation and blur the interpolated value with the values of very distant data points.

The interpolated surface goes through the data points. The value of the interpolation on a data point is equal to the value of that data point. This is not necessarily the case when using a function based interpolation system.

2.1 The Metasurface

At the time of this writing, AudioMulch's Metasurface could be the only preset interpolation system that features all these properties. Even so, it still has a few drawbacks. First, it is not possible to move the data points around while sound is generated. There is a mode in which the user can design the space (move the data points) and another one in which he can move the interpolation point (a performance mode). This is probably due to the fact that NNI requires a voronoi diagram to be calculated, a process that is computationally intensive[8].

The second drawback is the fact that the Metasurface is limited to two-to-many mappings. The goal of the work presented here is to allow many-to-many mappings using a novel controller, which means that higher dimensions interpolation was required.

Because NNI is a computationally intensive and complex method (especially at higher dimensions) a new interpolation method was developed. It has all of the desirable properties metioned above and is simple enough to be calculated in real time for musical purposes.

3. THE INTERSECTING N-SPHERES METHOD

3.1 2D surface interpolation

The `PresetInterpolator` object uses a novel interpolation method: the intersecting n-spheres method. Here is how it works for a two dimensional surface (Figures 1 and 2):

1. A circle is drawn around the interpolation point. Its radius equals the distance to the nearest data point. This circle is redrawn every time the interpolation point or a data point is moved.
2. Circles are drawn around each data point. The radii of these circles are equal to the distance to the nearest data point or the interpolation point, whichever is nearest. These circles are redrawn every time the interpolation point or a data point is moved.
3. The data points' circles that intersect the interpolation point's circle are considered neighbours and will influence the value of the interpolation point.
4. The value of the interpolation point is a weighted average of the value of its neighbours. The weight of each point is equal to the ratio $\frac{A_l}{A}$ where A_l is the circle-circle intersecting area and A is the area of the data point's circle. A_l can be obtained by equation 1 where R and r are the radii of the two circles and d is the distance from one centre to the other [13].

$$\begin{aligned} A_l =& r^2 \cos^{-1}\left(\frac{d^2+r^2-R^2}{2dr}\right) \\ &+ R^2 \cos^{-1}\left(\frac{d^2+R^2-r^2}{2dR}\right) \\ &- \frac{\sqrt{(-d+r+R)(d+r-R)(d-r+R)(d+r+R)}}{2} \end{aligned} \tag{1}$$

3.2 Extending the system to N dimensions

The ideal way of extending this method to three or more dimensions would be to modify equation 1 so that it yields a volume instead of an area and then, at step 4 of the method, to use a volume ratio instead of an area ratio. However,

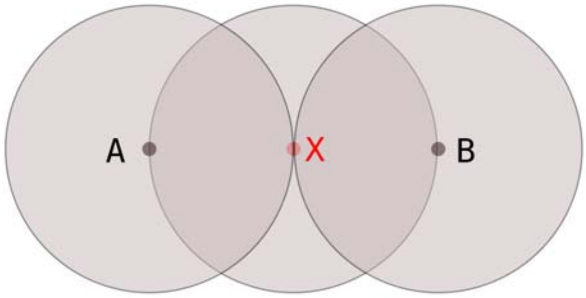

Figure 1: This preset interpolation space has two data points (A and B). X is the interpolation point. Circles are drawn around all the points. Both points are considered neighbours. Both points have the same weight because their ratios $\frac{A_l}{A}$ are equal.

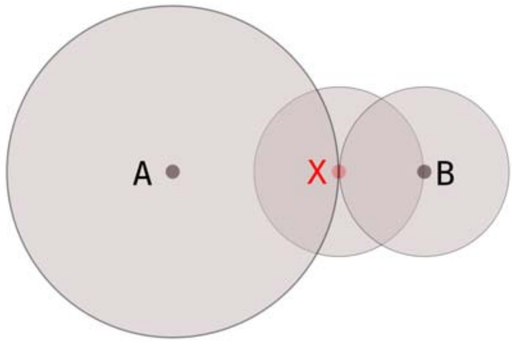

Figure 2: When X moves towards B, the weights of neighbors are adjusted. Both points are considered neighbors. Data point B will weight more than point A because its ratio $\frac{A_l}{A}$ is greater.

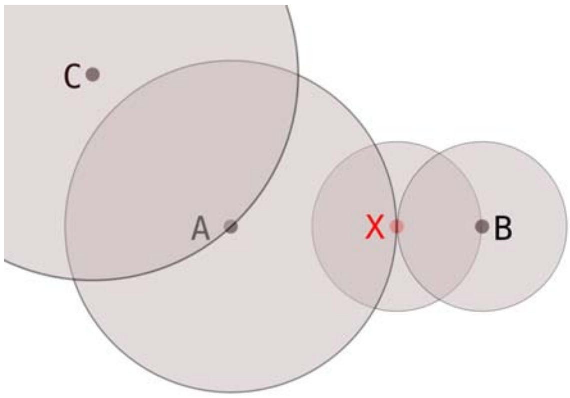

Figure 3: When X moves towards B, the weights of neighbours are adjusted. Points A and B are considered neighbours. Data point B will be given a greater weight than point A because its ratio $\frac{A_l}{A}$ is greater. C is not a neighbour and has no influence on the value of X.

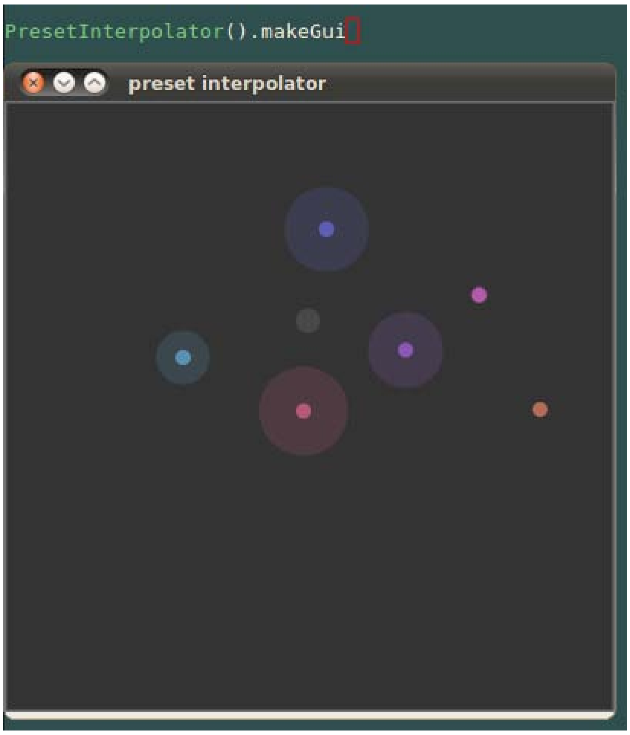

Figure 4: The `PresetInterpolator`'s GUI works only in two dimensions. The bigger grey dot is the interpolation point and the coloured dots are the data points. The translucent circles represent the weight of each of the data points.

this approach would render calculations much more complex and is beyond what is required for a creative or musical application.

In practice, it was found that a simple modification of the data structures so that they can represent n-dimensional data is sufficient. The equation 1 can be used without any modification: once the intersecting n-spheres are found, weights are obtained using area ratios.

While I admit that this is not theoretically correct, I found that the results of the interpolations are predictable and feel natural to the user. I also think that, in this context, the computational resources are better spent on audio synthesis or processing than on the interpolation method itself.

4. APPLICATIONS

4.1 Two dimensions

The 2D `PresetInterpolator` works a lot like the Metasurface: each preset (data point) can be edited and positioned using a graphical user interface (GUI) (Figure 3). It is possible to edit and navigate the space simultaneously without interrupting the sound, which is an improvement compared to the Metasurface. It can be used to control a process inside SuperCollider or to send MIDI or OSC messages to other applications. It offers all the usual editing capabilities: adding, removing and duplicating data points, as well as saving and recalling presets (spaces).

It has mostly been used to control granular and additive synthesis.

4.2 Using the sponge as a multidimensional space navigator

The sponge is a malleable musical interface that looks a lot like a cushion. It contains eight continuous sensors (two force sensing resistors and two 3D accelerometers) as well as 7 buttons [9]. From these sensor signals, more than 50 features (tilt, twist, fold, shocks, pressure, vibration, etc) are extracted and are available to be mapped to sound parameters.

The sponge has been used in conjunction with the `PresetInterpolator` to play live electronic music in concerts. The first attempts used only the two dimensional `PresetInterpolator` and therefore, only two of the features were used to control the `PresetInterpolator`. The result was satisfying: once the interpolation space was filled with data points corresponding to good sounding presets, it was possible to navigate intuitively and precisely in the space.

Next, many features of the sponge were mapped to the multidimensional `PresetInterpolator`. Up to eight features at the same time were used. In this context, the 2D GUI (Figure 3) was not suitable to represent a eight dimension space. In fact, the most convenient way to navigate and edit this space was to use the sponge itself. A new GUI was designed (Figure 4), but this time, it was much simpler: for each point in the preset space, there is a *A* (attach) button, a *E* (edit) button and a list of eight coordinates (the position of the point). To add a point in the space, the user can click a *Add Preset* button. He can then edit the preset (clicking on a *E* button opens a GUI that shows the values of every parameters of the preset).

The *A* button is used to attach a point to the cursor. This means that as long as the *A* button of a point is activated, this point will move with the cursor. This way, it is possible to position a point in the eight dimension space using the sponge. In other words, the user can explicitly associate a state (orientation and deformation) of the sponge to a sound (a preset).

The sponge requires both hands to be manipulated; it is therefore quite difficult to operate a mouse at the same time. For this reason, a button on the sponge is used to deactivate the *A* button.

5. CONCLUSION

The strategy presented here eases the design of many-to-many mappings. As such, it is comparable to simplicial interpolation [6] and mappings using neural networks or matrices [2], but it remains an explicitly defined mapping strategy.

The properties of the intersecting n-spheres interpolation method still have to be investigated and a comparison with other methods should be made. However, it was found that if the data points are distributed relatively uniformly, the interpolated surface will closely resemble the one obtained with the NNI method, while they can be considerably different when the data points density is irregular.

Nonetheless, the method has been used extensively by the author and the interpolated result feels natural and can definitely be used for musical purposes. The actual implementation in SuperCollider is stable but the performances get slow when there are many points in a preset space. An implementation in the C language is planned.

All the code is freely available on github.com/marierm and more information and videos are available on the author's web site: www.martinmarier.com.

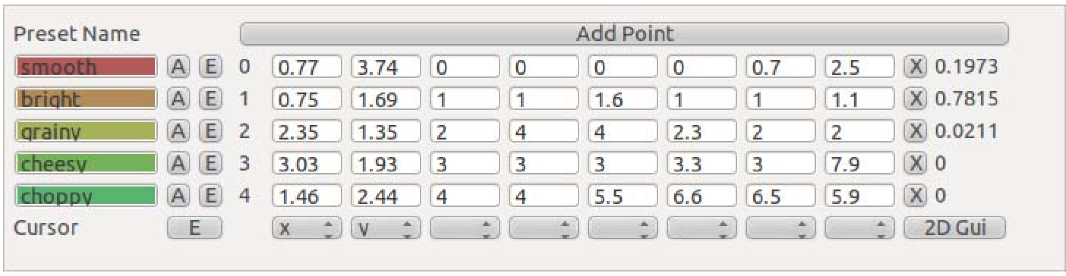

Figure 5: A simple GUI for the eight dimension preset interpolator. There is one line per point. Each point has a name, a *A* (attach) button, a *E* (edit) button, a *X* (delete) button. The value in the last column is the current weight of the point.

6. ACKNOWLEDGMENTS

The author is supported by the *Fonds québécois de la recherche sur la société et la culture* (FQRSC), the *Observatoire interdisciplinaire de création et de recherche en musique* (OICRM) and the Center for Interdiscplinary Research in Music, Media and Technology (CIRMMT). He is grateful to Dr. Garth Paine and Prof. Jean Piché.

7. REFERENCES

[1] R. Bencina. The metasurface: applying natural neighbour interpolation to two-to-many mapping. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 101–104. National University of Singapore, 2005.

[2] F. Bevilacqua, R. Müller, and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 85–88. National University of Singapore, 2005.

[3] J. Boissonnat. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry*, 22(1-3):185–203, May 2002.

[4] E. Favreau. Les outils de traitement GRM Tools. *Publications du LMA*, 1998.

[5] A. Freed, J. MacCallum, A. Schmeder, and D. Wessel. Visualizations and Interaction Strategies for Hybridization Interfaces. In *NIME '10: Proceedings of the 2010 conference on New interfaces for musical expression*, number Nime, pages 343–347, Sydney, 2010.

[6] C. Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(02):85–96, 2003.

[7] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.

[8] H. Ledoux and C. M. Gold. Spatial Interpolation : From Two to Three Dimensions. In *GIS Research UK, 13th Annual Conference*, pages 518–523, Glasgow, 2005. GIS Research Centre, School of Computing, University of Glamorgan.

[9] M. Marier. The Sponge: A Flexible Interface. In *NIME '10: Proceedings of the 2010 conference on New interfaces for musical expression*, pages 356–359, Sydney, 2010.

[10] A. Momeni and D. Wessel. Characterizing and controlling musical material intuitively with geometric models. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 54–62. National University of Singapore, 2003.

[11] G. Paine. Towards Unified Design Guidelines for New Interfaces for Musical Expression. *Organised Sound*, 14(02):142–155, 2009.

[12] M. Spain and R. Polfreman. Interpolator: a two-dimensional graphical interpolation system for the simultaneous control of digital signal processing parameters. *Organised Sound*, 6(02):147–151, Feb. 2002.

[13] E. W. Weisstein. Circle-Circle Intersection.