On the Choice of Mappings Based on Geometric Properties

Doug Van Nort, Marcelo M. Wanderley, Philippe Depalle
Signal Processing and Control Laboratory
Faculty of Music - McGill University
555 Sherbrooke St. West
Montreal, QC Canada H3A 1E3
doug@music.mcgill.ca

ABSTRACT

The choice of mapping strategies to effectively map controller variables to sound synthesis algorithms is examined. Specifically, we look at continuous mappings that have a geometric representation. Drawing from underlying mathematical theory, this paper presents a way to compare mapping strategies, with the goal of achieving an appropriate match between mapping and musical performance context. This method of comparison is applied to existing techniques, while a suggestion is offered on how to integrate and extend this work through a new implementation.

Keywords

Mapping, Interface Design, Interpolation, Computational Geometry

1. INTRODUCTION

There are currently a large number of physical controllers and hardware/software synthesis environments available for use in computer music[1]. As a whole, the area of gestural control has grown rapidly in the past several years. Only more recently[2], however, has attention been given to the process by which data from an input device is associated with synthesis parameters for real-time control. We refer to this process as *Mapping.*

1.1 Musical Use of Mappings

In considering a mapping strategy for a real-time performance system, a basis for comparison must be established to aid in the selection process. As explicit mappings are themselves functional expressions, their appropriateness for a particular musical use can partially be determined by the properties that these functions possess. In order to do this, one must first consider what a desirable property of a mapping may be in a musical context. These properties are determined by the desired perceptual effect in both gesture and sonic result, combined with the choice of controller.

In general there are certain properties which find importance in many musical situations. Perhaps the most obvious is *continuity.* It may also be essential that a rate of change of a gesture produce some result in a sound space, and so *differentiability*[3] may be desirable. Taking this further, we may require that this rate of change itself be continuous, or perhaps greater smoothness within a gesture is required so that we may need *continuous higher derivatives* to exist. In terms of flexibility, we may require *ease of editing* for moving between controllers or synthesis algorithms. Depending on one's resources, the *computational complexity* of a mapping scheme may also play a large role.

Regardless of the performance situation, it is important to remember that we need to discretize any data that we wish to store on a computer. This implies that any mapping which we implement will need to perform interpolation or approximation. The necessity of exactness via interpolation depends on several things, including quantization, resolution of transducers and perceptual discrimination of gesture and resultant sound. As we will see, there is a tradeoff between exactness, smoothness and computability of a mapping.

1.2 Mathematical Formalization

Mathematically speaking, a mapping $f : A \to B$ is a function between two sets such that for every $a \in A$, there exists a unique element $f(a) \in B$. Notice that this definition allows for a mapping in which several elements $a_1, \dots, a_n \in A$ correspond to the same element $b \in B$ (that is, $f(a_1) = \cdots = f(a_n) = b$), or for a one-to-one relationship in which there exists a unique element in B for each member of A. In order for f to be well defined there cannot exist multiple elements $b_1, \dots, b_n \in B$ such that $f(a) = b_1 = \cdots = b_n$. However, the notion of a "one-to-many" mapping makes sense if we consider a function g which maps across dimensions of a multi-dimensional range. More specifically, a mapping $g : X \to Y$ is one-to-many if there exists

$$(x_1, x_2, \dots, x_n), (x'_1, x_2, \dots, x_n) \in X$$

such that

$$g((x_1, x_2, \dots, x_n)) = (y_1, y_2, \dots, y_m)$$

$$g((x'_1, x_2, \dots, x_n)) = (y'_1, y'_2, \dots, y'_m)$$

with $y_i \neq y'_i$ for 2 or more variables y_i. In other words, varying only a single parameter in X affects change in more than one dimension of Y.

In a musical performance context, we may consider a set of controller parameters X and synthesis parameters Y to be subsets of the vector spaces $\mathbf{R}^n$ and $\mathbf{R}^m$ respectively, which allows for a geometric representation. The formal expression of this then becomes:

$$X = \{(x_1, \dots, x_n) | x_i \in \mathbf{P}_i \subset \mathbf{R}, i = 1, \dots, n\}$$

[1]The reader is directed to [14] for examples of existing controllers which have found musical use.
[2]See [15] for a thorough discussion of recent work.
[3]For the essential mathematical definitions needed for this article the reader is directed to http://www.music.mcgill.ca/musictech/spcl/mapping.html

where each i corresponds to a different parameter and $\mathbf{P}_i$ is the subset of $\mathbf{R}$ that contains the possible values of parameter i. With this representation, each element of X is an n-dimensional vector and each coordinate represents a single control parameter. We assume from this point forward that $X \subset \mathbf{R}^n$ is our control parameter space and $Y \subset \mathbf{R}^m$ is our space of sound synthesis parameters.

We can find an analogy to this in the musical use of mapping in [6], in which the authors define 3 subcategories[4]:

•one-to-one[5]: one input parameter is mapped to one synthesis parameter

•one-to-many: one input parameter is mapped to several synthesis parameters

•many-to-one: many input parameters are mapped to a single synthesis parameter.

The approach that we take defines X and Y to be subspaces of a (potentially) high-dimensional Euclidean space. This allows for a geometric representation that can prove beneficial when thinking about the proper choice of mapping, as the geometric properties that a mapping possesses can give insights into its ability to transition between gestural and/or sonic dimensions. As we consider here mappings of a continuous surface, we can assume that any reference to controllers refers to those that produce continuous variables. Another presupposition to this discussion is that our mappings are *explicit*. That is, mappings with an analytic expression that is known to the interface designer [6].

2. MAPPING STRATEGIES

2.1 Some Existing Mappings

Several works have considered the notion of multiple layers of mapping, specifically [1], [9] and [13]. In [13] the authors consider an "abstract layer" Z of d-dimensions that exists between controller space X and synthesis space Y. The first mapping is described as an adapter between X and the abstract parameters. The points which are known in sound space are stored in a d-dimensional lattice. The second mapping, between the abstract layer and Y, is a multilinear interpolation based on the 2^d points in the hypercell that contain the input point. This method is continuous and is differentiable, but does not have a continuous derivative (this being discontinous at the join between hypercells). The computation for this method increases exponentially with the dimension of the space within which it is embedded, in this case Z, and in particular is $O(d2^d)$.

In [2] a mapping $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ is defined which also interpolates points spaced in a grid. As with the last, the known control points of this method need to conform to a shape which is topologically equivalent to an n-dimensional lattice. These control points are associated with points in synthesis space Y. This gives us an exact, pointwise mapping between some collection of points in controller space and some corresponding points in synthesis space - there is no intermediate layer as in the previous example. Geometrically speaking, this amounts to our n-dimensional lattice being embedded in $\mathbf{R}^m$. To interpolate to other points within the grid, a scheme is developed which first partitions each hypercell into simplices. When a point is input into the lattice (when the controller outputs a parameter stream), the algorithm determines which subdivision the point lies in. If the point p does not lie at a vertex of the lattice (a known data point), the vertices $\{v_0, \ldots, v_n\}$ of the simplex which contains p determine the point as follows:

$$p = v_0 + \sum_{i=1}^{n} \alpha_i v_i, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \in \mathbf{R}\ \forall i$$

In other words the point p is expressed in terms of barycentric coordinates with respect to its containing simplex. This scheme is piecewise linear, and it requires that control points are spaced relative to some grid. Therefore, in terms of space requirements this technique is the same as the previous, with a storage requirement determined by the 2^n vertices of a hypercell multiplied by the number of such cells needed to cover a space. However, this scheme differs from multilinear interpolation in that it reduces the number of necessary operations[7] from $O(n2^n)$ to $O(mn)$. Further, this method is non-differentiable at the edges between subdivisions[8]. That is, there are sharp creases at the join between simplices. This difference in the smoothness of these two mappings is clear when we consider that the representations of the two are fundamentally different - the simplex based scheme is comprised of "flat" sections, while multilinear interpolation has a curved contour, which in fact is hyperbolic.

Another mapping technique that utilizes simplex based interpolation is presented in [5]. This method, referred to as "simplicial interpolation" extends the results from [2] in several ways. As with this preceding work, it is based on a pointwise mapping between $X \subset \mathbf{R}^n$ and $Y \subset \mathbf{R}^m$. The difference between the two is that this approach allows the collection of known points to be scattered. The author achieves this by creating a triangulation of the points in $\mathbf{R}^n$ rather than fitting them to a grid. In particular, the method used to do this is the *Delaunay triangulation*, which has been widely used for spatial interpolation.

Now, this triangulation in $\mathbf{R}^n$ induces a similar one in $\mathbf{R}^m$, giving us a mesh embedded in the higher-dimensional space in analogy to the embedded lattice in [2]. As in the previous method, the point's barycentric coordinates are determined, which interpolates the surface defined by the triangular mesh in X. This likewise interpolates the surface induced in Y. Depicted[9] in figure 1 is a triangulation of a two-dimensional controller space, which induces an embedding of "crinkled" two-dimensional triangles in a three-dimensional sound synthesis space. It is one-to-one in regards to mapping of simplices, but notice that it is not itself a one-to-one mapping. Rather, it is potentially a many-to-many mapping, depending on the orientation of each simplex in space. Further, it allows for scattered data points, can be edited

[4]A similar categorization of mapping types was also made in [11] using different language (one-to-one, convergent, divergent).

[5]There is a general consensus that the strict use of one-to-one mappings in an instrument design leads to a lack of expressive potential. See [6] for a discussion of this.

[6]This is in contrast to *implicit* mappings, which rely on internal adaption of a system. See [6] for further discussion on this distinction.

[7]It is claimed in [5] that the actual time to compute is $O(n^3 + mn)$, the same as for the Simplicial Interpolation method.

[8]It is suggested in [3] that this be dealt with by introducing B-spline blending functions. See [4] for more on this technique.

[9]Figure 1 consists of screen shots of C++ implementation of simplicial interpolation by Camille Goudeseune, with modification by Doug Van Nort.

property	Multilinear([13])	Simplex Scheme 1([2])	Simplex Scheme 2([5])	RST([7])
continuous	yes	yes	yes	yes
local diff	yes	yes	yes	yes
global diff	yes	no	no	yes
C^k	k = 0	k=0	k=0	variable
grid vs. scattered	grid	grid	scattered	scattered
adjustable smoothing	no	no	no	yes
complexity rank(1 = fastest)	3	1	2	4

Table 1: Mapping Strategies and Certain Mathematical Properties

locally, and the number of points is not constrained by the structure of a grid, making this technique more flexible than [2] in general.

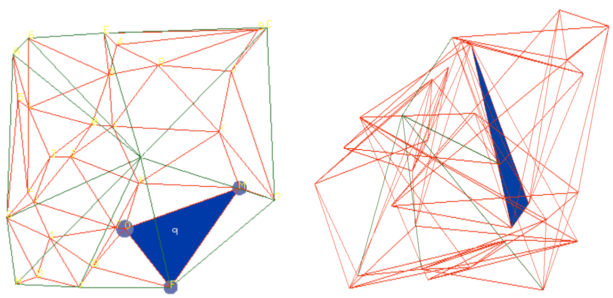

Figure 1: Example of Mapping from [5]. Left: Controller Space in 2D Right: Synthesis Space in 3D

There are many similarities between [2] and [5]. Both are C^0, piecewise linear and are non-differentiable at the edges between simplices. Thus, the former technique may be appropriate for data points which lie in a grid. However, the Delaunay triangulation has several properties that make it a desireable choice, including regularity of angles (less singularly sharp points) and the ability to give a best approximation of certain smoother functions (see [10] for more on this). Thus, for sufficiently large number of points this approach may give an acceptable approximation of a smooth surface while keeping computational load relatively light. Naturally, there is a tradeoff in this case with less stored values corresponding to a rougher approximation of differentiability.

2.2 Combinations and Extensions

The aforementioned mappings, all of which are continuous and explicitly defined, have different attributes which make them appropriate for different musical contexts. We have seen that each one has qualities which make it an appropriate choice for certain musical situations, but that a tradeoff likely exists. To this end, we propose the use of a mapping strategy which combines several of the qualities of those from the previous section, making it a potentially powerful method. The technique, known as *Regularized Spline with Tension* (RST), is presented in theory and in application in [7] and [8], respectively. This is a radial basis function approach that computes coefficients based on a matrix of linearly indepedent equations of stored data points. For the purpose of real-time control, these coefficients are pre-computed. As with other spline-based approaches, unwanted overshoots can occur due to fluctuations of data points and sparse data sets. The authors of [7] alleviate this problem by a modification of the standard method to include free parameters that adjust smoothness and tension, essentially tuning the effect of higher derivatives. The varying of these parameters allows the mapping to move between rigid and flexible models, while varying the amount of approximation.

Not only is this technique continuous and smooth, but it may be constructed so that it is C^∞, or in other words we can make it as smooth as is necessary. The ability to tune the mapping to relative levels of approximation is beneficial when working with different data sets possessing varying amounts of noise.

This technique encompasses many of the desirable properties of the previous results, as can be seen in table 1. With adjustable smoothness and tension, it also has potential for great flexibility. Quite naturally, this comes at a cost in terms of computational complexity. However, RST has been implemented efficiently for two and three dimensions as part of GRASS GIS, or Geographic Resources Analysis Support System - an open source geographic information system. [10] Further, we have implemented a real-time version of RST in Max/Msp.

2.3 Dimensionality

Control of sound synthesis often requires a mapping to go from lower to higher-dimensional space. The simplex-based method of [5] is designed to reduce the dimensionality of points in synthesis space Y to that of controller space X by directly associating them in a pointwise manner. This is achieved by using a generative technique (Sammon's mapping or a genetic algorithm) to preserve the clustering properties of the points in synthesis space and map them into the lower-dimensional space. However, there still needs to be some decision as to why certain points map onto others in the lower-dimensional surface, one that takes into account the relevance of gesture and resulting sound. The author also suggests that this be done by the user in an intuitive manner, associating certain controller space points with certain points in synthesis space, in a sense creating a "user-defined perceptual subspace". Dimensionality reduction may be introduced in conjunction with the other mappings as well, using such techniques as multi-dimensional scaling (MDS), principle component analysis (PCA) and self-organizing maps (SOM). In doing this the user must consider the sort of subspace that needs to be constructed, and whether it is perceptually driven or data driven. An example of a framework that generates a controllable sub-

[10] Work in this field explicitly deals with the mapping of geographic data for visual display, and so spatial interpolation plays a major role.

space based on input data and interpolates points within this space can be found in [12]. This work presents a performance system in which continuous physical controller data drives the formation of an input-output space via cluster-weighted modeling. However, it does not focus specifically on the continuous mapping strategies involved.

The interest here is precisely on these mappings. Geometrically speaking, our controller space X is bound by the dimensionality of the input device - the high-dimensional interpolator of [5] embedded in $\mathbf{R}^m$ still consists of piecewise linear n-dimensional surfaces when mapping a device of n dimensions. Similarly, methods such as multilinear or regularized spline with tension generate an n-dimensional surface[11], which can then be embedded in higher-dimensional space.

3. EXAMPLE

In order to focus on the effect of the mapping, a simple low-dimensional example was preferred to a more complex performance system. The goal is to illustrate the importance of the proper choice of mapping given the perceptual nature of the musical task at hand, while still providing an example that is intuitive and relevant in a musical context. In this example, we used the x-y position of a wacom tablet to control a simple source-filter system in which an harmonic signal of fundamental frequency f_0 is filtered by a second order IIR filter of the form

$$y[n] = x[n] + b_1 y[n-1] + b_2 y[n-2]$$

This $\mathbf{R}^2 \rightarrow \mathbf{R}^3$ mapping is achieved by embedding the wacom control surface in the synthesis space, so that x-y motion controls the three variables f_0, b_1, b_2. More specifically, controlling the filter bandwidth and center frequency allows the b_1 and b_2 coefficients to be controlled indirectly by the relationship

$$b_1 = 2e^{-\pi B_w / f_s} cos(2\pi f_c), b_2 = -e^{-2\pi B_w / f_s}$$

where B_w, f_s, and f_c represent bandwidth, sampling rate and center frequency. While the filter coefficients are the target space, the embedding is in the more perceptually relevant space of spectral bandwidth and center frequency. Thus, the geometry of the mapping - its orientation and embedding in a higher-dimensional space - allows for a control similar to that of multiple layers of mapping, as can be seen in the mapping flow diagram of figure 2.

The control surface is 2-dimensional with the x position of the tablet controlling filter bandwidth in the range of 1-100 Hertz and the y position controlling the fundamental of the harmonic source in the range 500-2000 Hertz. It is embedded such that the x-y position across the surface is mapped to center frequency of the filter, which ranges from 2 to 8 times the frequency of the given fundamental. Thus, we set up an example musical situation in which slow, smooth gestures can be used to "explore" the sonic space in order to scan the harmonic peaks that occur at the point of resonance f_c. Changing only the mapping allows us to hear the effect that it has on the feel of the interface[12].

[11]Assuming the mapping is directly from controller space in $\mathbf{R}^n$, and there is no intermediate layer of another dimension as in [13].

[12]A study in which the authors tested the effect that different mappings have on expressivity was reported in [11].

Given the sensitivity of human perception in the chosen frequency range and the sort of slow "exploratory" gesture we were after, we chose to use the two methods that are globally differentiable - namely multilinear interpolation (in this case bilinear) and RST. Recall that the former is a grid-based technique, and so for purposes of comparison we used the same gridded points for both.

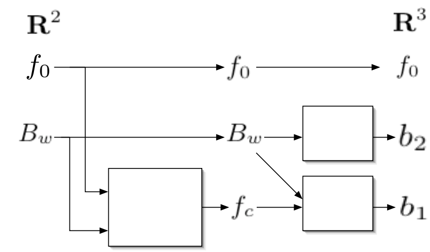

Figure 2: Effect of Embedded Surface on Parameter Control

The tests were carried out in Max/Msp, in which we have written multilinear and RST objects. The evenly spaced dots in Figure 3 depict the data points that were stored in a 7x7 grid across the full range of possible tablet values, with the borders of the tablet corresponding to a center frequency of $2f_0$ and the very center of the tablet corresponding to $f_c = 8f_0$.

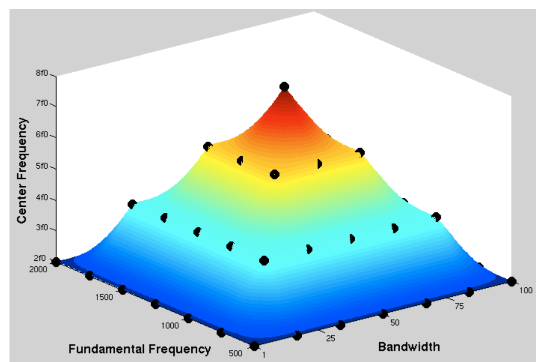

Figure 3: Bilinear Interpolation and Stored Data Points

As the multilinear scheme is exact, the interpolated values were distributed montonically from edge to center, with a sharp crease at the midpoint and at the diagonals to form a pyramid shape, as can be seen from figure 3. In contrast, the RST mapping provided approximate values, and so it gave its own global maximum which was slightly off from the midpoint. In this musical context, relative as opposed to absolute positioning is more important, and so the exactness of point locations is not of great concern. As can happen with spline-base techniques, the initial RST mapping produced many overshoots in value. Using the smoothing and tension parameters, however, we were able to generate a mapped surface with minima at the edges, a maxima near the midpoint, and a smooth contour. This smoothness, and the contrast to the bilinear interpolant can be seen in figure 4, which depicts a 2-dimensional plot of mappings for the given data points, with a color mapping for the third dimension. This ability to adjust the mapping contour is one of the main advantages of the technique.

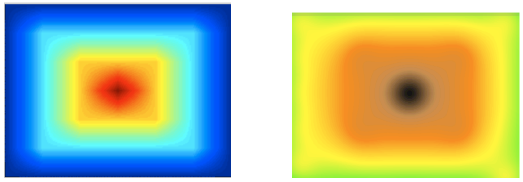

Figure 4: left: Bilinear Surface right: RST Surface

In testing these methods there were two differences between them that were apparent. The first was behavior at the maximum f_c value. As the multilinear technique had a sharp crease at this point, and as the resonant filter had a narrow bandwidth, the interpolation produced offset values at this singular point which caused the frequency value to jump, thereby losing the strong eighth harmonic presence. This did not occur with RST, as there were more values in the proximity of the maximum which diminished the effect of offset errors. The second difference, which affected the feel of the interface, was the transition into and out of points of resonance. As the IIR filter used had a narrow band and sharp resonance, the frequency range was small and the spectral peaks were sharp. However, using the multilinear mapping made it difficult to find the harmonic peaks until they were passed over, causing sharp jumps in energy at the given harmonic. In contrast, the smoothness of the RST mapping made it possible to hear when peaks were close by, allowing more precise control of the sound by moving around or through these points as desired.

This example illustrates how the context determined the proper mapping. The gestural and sonic perceptual nature of the situation called for both local and globally smooth changes of relative position. Therefore, the RST technique proved to be better a choice. In situations where discontinuity exists in the signal itself, or where "glitches" in the sound are desired for aesthetic reasons, another method may be more appropriate. In the latter case mappings may be combined so as to provide a "texture map" to a musical control surface.

4. CONCLUSIONS

It is hoped that this paper will facilitate the discussion on the issue of mapping between controller and sound synthesis parameters. By introducing mathematical formalisms, functional properties that are relevant in general to computer music performance may be considered. Doing this provides a basis for comparison, which we have done here in the case of explicit continuous mappings. The goal is not to suggest one method over another categorically, but rather to illustrate the importance of the proper selection of mapping. With this in mind, RST was implemented as it combines several of the useful properties that we have identified, and thus offers the possibility of greater flexibility for many real-time situations. The example provided illustrates how both the feel and functionality of an interface can be affected by mappings that have subtle mathematical differences.

5. ACKNOWLEDGMENTS

Special thanks to Camille Goudeseune for several helpful discussion and for assistance in implementing his simplicial interpolation code.

6. REFERENCES

[1] D. Arfib, J. Couturier, L. Kessous, and V. Verfaille. Strategies of Mapping Between Gesture Data and Synthesis Model Parameters Using Perceptual Spaces. *Organised Sound*, 7(2):127–144, 2002.

[2] I. Bowler, A. Purvis, N. Bailey, and P. Manning. On Mapping N Articulation onto M Synthesiser-Control Parameters. In *Proc. of the 1990 International Computer Music Conference*, pages 181–184, 1990.

[3] I. Choi, R. Bargar, and C. Goudeseune. A Manifold Interface for a High Dimensional Control Space. In *Proc. of the 1995 International Computer Music Conference*, pages 181–184, 1995.

[4] G. Farin. *Nurbs: From Projective Geometry to Practical Use.* A.K. Peters, Ltd., Natick, Massachusetts, 1999.

[5] C. Goudeseune. Interpolated Mappings for Musical Instruments. *Organised Sound*, 7(2):85–96, 2002.

[6] A. Hunt and M. Wanderley. Mapping Performance Parameters to Synthesis Engines. *Organised Sound*, 7(2):97–108, 2002.

[7] H. Mitasova and L. Mitas. Interpolation by Regularized Spline with Tension: I. Theory and Implementation. *Mathematical Geology*, 25(6):641–655, 1993.

[8] H. Mitasova and L. Mitas. Interpolation by Regularized Spline with Tension: II. Application to Terrain Modeling and Surface Geometry Analysis. *Mathematical Geology*, 25(6):657–669, 1993.

[9] A. Mulder and S. Fels. Sound Sculpting: Manipulating Sound through Virtual Sculpting. In *Proceedings of the Fifth Brazilian Symposium on Computer Music*, pages 151–164, 1998.

[10] S. Omohundro. The Delaunay Triangulation and Function Learning. *Intl. Computer Science Institute Tech Report*, pages 1–10, 1990.

[11] J. Rovan, M. Wanderley, S. Dubnov, and P. Depalle. Instrumental Gestural Mapping Strategies as Expressive Determinants in Computer Music Performance. In *Proc. of Kansei, The Technology of Emotion Workshop*, pages 68–73, 1997.

[12] B. Schoner, C. Cooper, C. Douglas, and N. Gershenfeld. Data-Driven Modeling of Acoustical Instruments. *Journal of New Music Research*, 28(2):81–89, 1999.

[13] M. Wanderley, N. Schnell, and J. Rovan. Escher - Modeling and Performing Composed Instruments in Real Time. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, pages 1040–1044, 1998.

[14] M. Wanderley-editor. *Trends In Gestural Control of Music.* IRCAM – Centre Pompidou, 2000.

[15] M. Wanderley-editor. Mapping Strategies in Real-Time Computer Music. *Organised Sound*, 7(2), 2002.