PARAMETERIZED MORPHING AS A MAPPING TECHNIQUE FOR SOUND SYNTHESIS

Chinmay Pendharkar, Michael Gurevich, Lonce Wyse

Mixed Media Modeling Lab
Institute of Infocomm Research, Singapore
lonce@i2r.a-star.edu.sg

ABSTRACT

We present a novel mapping technique for sound synthesis. The technique extends the familiar concept of morphing to the domain of synthesis parameters. A morph between defined points in the parameter space representing desirable sounds is itself parameterized with high-level controls. The choice of end points of the morph and the extent of the morph are used as input handles to map arbitrary control signals to the synthesis parameters. Additional off-line methods control the interpolation functions and selection of parameter points. We discuss a tool to allow creation, manipulation and usage of such mappings.

1. INTRODUCTION

Real-time interactive digital sound synthesis is becoming increasingly important in many applications such as music, games and personal electronics. Improvements in synthesis techniques and processing power, along with increased potential for expressivity make sound synthesis an attractive alternative to sample-based systems. A generic real-time interactive digital sound synthesis system can be modeled as a bank of synthesizers that expose certain parameters to be a controlled. A hardware or software controller changes the parameters to obtain the intended sounds. The controller generates a number of digital control signals as input to the system, which must be mapped to the parametric control handles of the synthesis models.

This mapping is an important part of the synthesis system. Inadequate mappings can cause even rich models capable of dynamic, responsive sounds, to sound dull and "mechanical". There remains a great deal of research of mapping techniques that allow synthesis models to give more expressive- or natural-sounding output, or to more substantially define the behavior of the system.

1.1. Related Works

Approaches to control in the form of mapping have existed in fields such as control systems for some time [1]. Only since the relatively recent advent of practical, parametric real-time digital synthesis has it become applicable to sound synthesis.

Over the years, many strategies have been proposed to approach mapping. A popular mapping strategy is to define formal deterministic relationships between control and synthesis parameters. These strategies are mainly based on techniques predominant in control theory, like linear algebra and matrix transformations [2].

Some authors choose to take non-linear and heuristic-driven approaches that lead to practical and interesting mapping strategies. These include using spatial layout of mapping or weighing functions over a representation of the input space [3] and the uses of geometric shapes to define input parameter spaces and mapping such shapes to shapes of higher dimensionality [4].

Some of these systems introduce one or more intermediate or "middle" mapping layers between control and synthesis parameters. In such schemes, control parameters are mapped onto intermediate parameters, which are in turn mapped to synthesis parameters. These intermediate parameters may be arbitrarily defined, may describe some higher-level (*i.e.* perceptual) features of the desired sound [5, 6], or may represent some virtual system whose features are mapped onto the synthesis parameters. For example, Schatter et al. [7] use a gestural controller to manipulate virtual graphics objects, whose features are mapped to synthesis controls.

Multiple layered mapping can allow the intermediate mappings to be reused. The mapping from the middle layer to the synthesis parameters may remain unchanged, while another controller is mapped to the middle layer [8]. Multi-layered systems are also one way of dealing with the problem of dimensional asymmetry. Dimensional asymmetry occurs when the dimension of the input control parameters and the output synthesis parameters is not equal. As such, mapping can be considered as a dimension reduction problem, and certain dimension reduction schemes, like simplical interpolation [9], can be used as a mapping strategy.

Though an m-input n-output configurable mapping technique is very generic and widely useful, a mapping technique with restricted input dimensions can be practical. Gestural control systems often have a limited number of controls with respect to the number of synthesis parameters. Such low input dimension mapping techniques allow users to comprehend and explore the mappings in much greater extent. Low-dimensional gestural controls can be more intuitive for the user and thus can lead to more expressive mappings. One example of this is a mapping strategy proposed by Bencina [10]. In the strategy the inputs are restricted to a 2-dimensional space. However no restriction is placed on the number dimensions of the output.

2. EVALUATING MAPPINGS

We define three important criteria for analyzing mappings: complexity, expressivity, and intuitiveness. There is certainly a strong relationship between a mapping technique and the degrees to which these properties are reflected in the mappings that are generated. One could conceive of quantitative metrics for assessing these, but this is beyond scope of this paper. We leave these as subjective assessments in order to create a vocabulary for discussing and informing the design of mapping techniques.

The complexity, especially the computational complexity of mappings, is an important factor when designing mapping for real-time audio synthesis engine. Complexity can be conceived as related to the number of computational or logical steps required to generate a synthesis parameter from an input control change. Expressivity refers to the ability of the user to generate a rich variety of sonic output from the synthesizer, under deliberate control. Mappings resulting in output that, on one hand "always sounds the same" or on the other hand are "random" or "uncontrollable" from the user's perspective are not expressive. Expressivity is not a measurable property of the mapping, but it's an affordance the mapping can provide. Exceptionally talented performers can make poor mappings expressive. However, by expressive mappings we mean those that relatively easily enable a broad range of expression. Intuitiveness in a mapping can be seen as related to the user's mental model of the mapping [11]. If the user requires only a simple conception of how the mapping works, then it would be considered intuitive.

The goal in creating a successful mapping strategy is to achieve an optimal combination of the three criteria. A good mapping strategy should be intuitive and produce expressive mappings, with the least amount of complexity possible in the mapping, but no less.

3. PARAMETERIZED MORPHING

Morphing is a process used widely in image processing. It is used in animations and motion pictures to change from one image to another through a seamless transition. This is generally achieved by interpolating between certain common features in the initial and final images.

Audio morphing normally describes the generation of a transition between two segments of recorded or synthesized audio. As in image morphing, it is normally achieved by interpolating between sets of features that are determined after the sound has been rendered, based on analysis of the audio signal in the time, frequency, or perceptual domains. For example, Slaney [12] describes a technique for morphing between sounds by interpolating between extracted spectral shapes and pitch.

3.1. Morphing in Parameter Space

With parametric, real-time synthesis, one can appropriate the term morphing and extend its application to interpolation in the domain of the synthesis parameters themselves, since they are available at the time of sound generation [13]. If for a particular synthesis model, sound A can be specified by a vector of parameters $\mathbf{P}_A$, and sound B can be specified by a vector of parameters $\mathbf{P}_B$, then we can generate a morph between A and B by interpolating between $\mathbf{P}_A$ and $\mathbf{P}_B$. Of course, the extent to which this creates a perceptual morph depends on the nature of the model and on the interpolation functions. For many classes of models, with appropriate interpolating functions, this can be an effective way of navigating sound spaces.

3.2. Mapping by Parameterized Morphing

We propose to use this concept of parameter-space morphing in the context of mapping. The basic mapping scheme is as follows. For any synthesis model, sets of parameters are defined in advance, corresponding to particular desirable or interesting points in the model's synthesis space. A pair of parameter sets can be chosen and set to be the end points for the morph. The morphing control would then choose the extent to which each of the parameter sets contribute to the output of the mapping.

This is achieved with some interpolating function used over each common parameter pair in the parameter sets. The user can also set the parameters at the end point dynamically, thus exploring the synthesis space.

This scheme leads to a 3-n mapping, where the mapping takes in the 2 discrete control inputs, and a continuous control inputs, and yields any number of continuous synthesis parameter outputs. The 2 discrete inputs are used to choose which parameter sets should be the end points of the morph, by indexing them from a list. The continuous input, a slider, could then determine the extent of the morph. See Figure 1 for a conceptual representation of parameterised morphing.

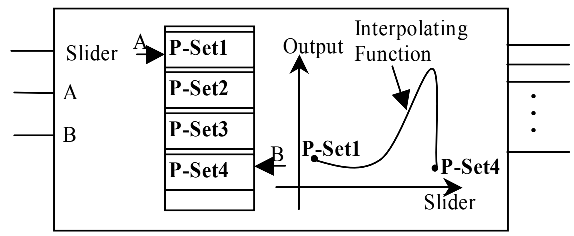

Figure 1: *Conceptual representation of parameterized morphing.*

3.3. Interpolation

A simple linear interpolation scheme will reduce the mapping to a simple range modification function and thus limit the utility and expressivity of the mapping. For example, while synthesizing the sound of a helicopter, if the desired control parameter of the mapping is the distance of the listener from the helicopter, the relationship between the input and the volume parameter of the model would be following the inverse square law. If the interpolation is limited to simple linear schemes, such common scenarios would not be addressable. Thus, a need for non-linear interpolation schemes arises. Piece-wise interpolation functions should also be available for dealing with discontinuous synthesis parameter spaces.

An interpolation scheme in morphing decides the output value of a certain feature, or parameter in sound synthesis, according to the two values being interpolated between. In the general case, the mathematical expression for the interpolation would be.

$$\mathbf{P}_O = (1 - \eta(s)) \cdot \mathbf{P}_A + \eta(s) \cdot \mathbf{P}_B \tag{1}$$

where $\eta(s)$ is the non-linear weighting function, $\mathbf{P}_A$ & $\mathbf{P}_B$ are the two parameter sets to be morphed between and s is the normalised value of the slider and $\mathbf{P}_O$ the output of the morph.

4. ADDITIONAL FEATURES

With this morphing scheme as a basis, we propose a series of extensions in order to design a usable mapping software system which attempts to satisfy our criteria of complexity, intuitiveness and expressivity. The objective is to create a mapping application that can operate with a variety of synthesis systems and controllers (physical or virtual). However, the large numbers of available synthesis systems and controllers present a challenge to design a single application that would offer universal connectivity.

We therefore designed a general-purpose mapping application, around which different interfaces can be defined, both for incoming control signals and outgoing synthesis parameters. The front-end interface can translate the control signals to the appropriate internal representation of the mapping system; we can call this a premapping. Similarly, the back-end interface translates the internal representation of the generated synthesis parameters into a protocol and format supported by the receiving application. See Section 5 for further details of the implementation.

4.1. Partitioning and Maplets

An extension to the morphing scheme arose from the need to have more control over the mapping. If all the synthesis parameters are considered to span an n-dimensional space, then each parameter set of length m leaves $(n - m)$ parameters undefined. Morphing between two such parameter sets, the mapping does not have any control over the remaining $(n - m)$ parameters. Furthermore, this scheme cannot produce divergent mappings as defined by Rovan *et al.* [14].

To address both these limitations, we introduce the concept of maplets. Maplets are objects, which contain the ability to do a single parameterized morph as discussed above. A complete map can contain one or more maplets.

Maplets are based on the idea of partitioning the set of all output parameter into groups of parameters that are related, especially in their dynamics with respect to an input parameter. For example, all parameters which should vary exponentially with respect to a single control may be grouped together. Such groups can be assigned to individual maplets, thus allowing each group to have independent morphing functionality and inputs. Maplets serve as the atomic mapping entity.

Furthermore, maplets allow for multiple pairs of parameter sets to be morphed simultaneously. This is useful when controlling complex sound models with multiple sets of parameters closely related in their behaviour. This feature is also needed when different interpolation functions are needed to map different groups of parameters. For example, when controlling a sound model of a car, it would be effective to control all the parameter related to the tyres separately from all the parameters related to the engine.

This scheme represents a $(2 + x) \longrightarrow n$ mapping, where x is any integer. The scheme need at least 1 continuous and 1 discrete control parameter, since Indices A and B could practically be premapped to 1 discrete input, for the lowest dimension case. Furthermore, multiple maplets can be manipulated by the same control parameter, thus allowing multiple sets of synthesis parameters to vary differently over a given input. Divergent mappings can now be generated by this scheme [14].

5. IMPLEMENTATION

We developed a tool to build and use mappings using parameterized morphing. The aim was to have a user friendly and intuitive means for designing mappings, and to test the practicality and verify the concept of the mapping scheme. Java was chosen as the development language for this project. Its mobility and portability have been incentives for its use in the development of a wide variety of applications for servers, PCs and mobile devices.

Modularity is important in the design of the system, since many of the subsystems are individual components that could be used on their own in a variety of mapping-related applications. The core mapping functionalities are separated from the communication subsystem and the GUI. See Figure 2.

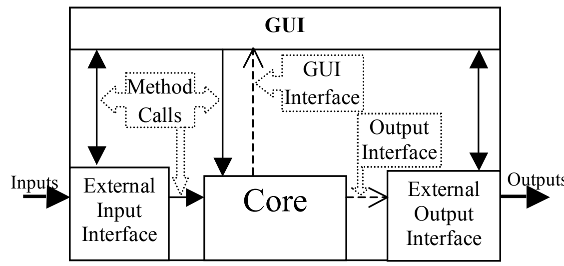

Figure 2: *Structure of the mapping tool.*

The core mapping functionality contains the interpolating and morphing logic of the mapping. It contains the maplets, the predefined parameter sets and the interpolation functions accessed through method calls.

The communication subsystem is based on Java Interfaces, which are provided in the core. Relevant communication systems can be designed according to the type of communication required by the application. Such a system can then be attached to the core using the interfaces provided. A default communication system has been implemented using the OpenSound Control (OSC) [15] protocol.

The GUI allows the tool to be used interactively. It provides functionality to audition the mappings that are created and attach interfaces to the mapping. It also gives the user a view of the current state of the system and the ability to store and load maps dynamically.

6. DISCUSSION

This implementation represents a multi-layered strategy, in that the notion of parameter sets represents an intermediate construct between the input and output parameters. One feature of this, as Wanderley [8] highlights, is that the mappings from parameter sets to synthesis parameters can persist regardless of the control parameters. Parameter sets can be reused in different maps and maplets. From the user's perspective, the concept of parameter sets also simplifies the mental representation of a potentially large space of synthesis parameters. Once parameter sets representing desirable points in the synthesis space have been defined in an off-line process, the user can ignore the target parameters themselves, and rather conceive of two sounds and a morph between them.

Mappings can be classified as one-to-one, convergent and divergent [14]. A good mapping scheme allows for all three types of mappings to be defined. The mapping scheme discussed allows oneto-one as well as divergent mapping. However, it is unable to produce a convergent mapping since, such a mapping does not fit easily into this morphing framework. If there is a need for simple convergent mappings to be coupled with this mapping scheme, they can easily be added as the pre-mapping layer to this mapping scheme.

7. FUTURE IMPROVEMENTS

There are a few areas for improvement that will be addressed in further work. These include further refinements to the specification of the interpolation function. A tool that would allow users to graphically specify the shape of the interpolator in real-time is currently under development. Currently, the user must manually specify the OSC interface, the default communication system. This will be extended with a tool that can automatically explore the OSC address spaces of both the controller and synthesizer. This would be a key application for the implementation of a proposed OSC query scheme [16].

Interfaces for protocols other than OSC, including MIDI, and a native interface to our own Java-based synthesis environment [17] also are under development. The source code of this project will be made available shortly, pending internal approval.

8. REFERENCES

[1] K. Ogata, *Modern Control Engineering*, 4th ed. Prentice Hall, 2001.

[2] F. Bevilacqua, R. Müller, and N. Schnell, "MnM: a Max/MSP mapping toolbox," in *Int. Conf. New Interf. for Musical Expr. (NIME'05),* Vancouver, Canada, 2005, pp. 85–88.

[3] A. Momeni and D. Wessel, "Characterizing and controlling musical material intuitively with geometric models," in *Int. Conf. New Interf. for Musical Expr. (NIME'03),* Montréal, Canada, 2003, pp. 54–62.

[4] D. van Nort, M. M. Wanderley, and P. Depalle, "On the choice of mappings based on geometric properties," in *Int. Conf. New Interf. for Musical Expr. (NIME'04),* Hamamatsu, Japan, 2004, pp. 87–91.

[5] D. Arfib, J. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model," *Organised Sound*, vol. 7, no. 2, pp. 122–144, 2002.

[6] A. Hunt and M. M. Wanderley, "Mapping performer parameters to synthesis engines," *Organised Sound*, vol. 7, no. 2, pp. 97–108, 2002.

[7] G. Schatter, E. Züger, and C. Nitschke, "Synaesthetic approach for a synthesizer interface based on genetic algorithms and fuzzy sets," in *Proc. Int. Comp. Music Conf. (ICMC'05),* Barcelona, Spain, 2005, pp. 664–667.

[8] M. M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proc. IEEE, Spec. Issue on Eng. and Music – Supervisory Control and Auditory Communication*, vol. 92, no. 4, pp. 632–644, 2004.

[9] C. Goudeseune, "Interpolated mappings for musical instruments," *Organised Sound*, vol. 7, no. 2, pp. 85–96, 2002.

[10] R. Bencina, "The Metasurface – applying natural neighbour interpolation to two-to-many mapping," in *Int. Conf. New Interf. for Musical Expr. (NIME'05),* Vancouver, Canada, 2005, pp. 101–104.

[11] D. A. Norman, *Some Observations on Mental Models*. D. Gentner and A. L. Stevens (eds.), Lawrence Erlbaum Associates, 1983.

[12] M. Slaney, M. Covell, and B. Lassiter, "Automatic audio morphing," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc.*, 1996, pp. 1001–1004.

[13] E. Altman and L. Wyse, *Emergent Semantics from Media Blending*. Srinivasan and Nepal (Eds.), The Idea Group Inc., 2004.

[14] J. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle, "Instrumental gestural mapping strategies as expressivity determinants in computer music performance," Synthesis Team/Real-Time Systems Group, Ircam, Paris, France, Tech. Rep., 1997.

[15] M. Wrigh and A. Freed, "Open SoundControl: A new protocol for communicating with sound synthesizers," in *Proc. Int. Comp. Music Conf. (ICMC'97),* Thessaloniki, Greece, 1997, pp. 101–104.

[16] A. Schmeder and M. Wright, "A query system for Open Sound Control," in *Proc. 2004 OSC Conf.*, Accessed Mar. 21, 2006, [Online] http://www.opensoundcontrol.org/proceedings.

[17] L. Wyse, "A sound modeling and synthesis system designed for maximum usability," in *Proc. Int. Comp. Music Conf. (ICMC'03),* Singapore, 2003, pp. 447–451.