Performance Factors in Control of High-Dimensional Spaces
Guy E. Garnett
Camille Goudeseune
University of Illinois at Urbana-Champaign
{garnett,cog}@uiuc.edu
Keywords: cognition, perception, control, multidimensionality

Abstract

Our work on instrument design and instrumental performance interfaces has led us to consider in detail the mappings from the performer’s gesture space to the listener’s perceptual space. The performer’s gesture space includes the performer’s conceptualization of the instrumental interface, from their physical gestures to the connection of these gestures with the resultant sounds. Since high-quality performance requires tremendous control of sonic detail, we also need to consider the kinds of sounds that may result and the qualities of variability available in those sounds. We address each of these problems through new techniques for creating mappings between controllers and the sonic results.

1. Design Problems in Electronic Instrument Performance

In this project we attempt to solve some of the major synthetic-instrument design problems not by proposing new physical interfaces or controllers, and not by creating new synthesis algorithms, but by improving tools for understanding and designing the mappings between the physical interfaces and the resultant sounds—regardless of which synthesis algorithms are used.

The problems we consider are two of the major difficulties facing designers of new electronic instruments: one, the performer’s conceptualization of the instrumental interface, broadly conceived as including everything from the performer’s physical gestures to the connection of these gestures with the resultant sounds. Two, the kinds of sounds that may result and the qualities of variability available in those sounds. We address each of these problems through new techniques for creating mappings between controllers and the sonic results.

The issues associated with the first problem are, first, how to take advantage of the many years of practice that allow the performer to think largely in terms of the sound to be produced, and only secondarily, in particular when practicing or when playing a difficult passage, in terms of the instrument (“am I using enough bow pressure here to get the sound I want?”) or even in terms of their own body (“maybe I should lean into this note more”). A part of this is in retraining reflexes and motor controls to do the right gesture at the right time; we call this physical retraining. Problems associated with physical retraining can be minimized by retaining a physical interface that is functionally very similar to the practiced instrument, such as a keyboard controller for pianists and, in our case, a violin controller for violinists.

However, there is a more substantial issue involved that has received less attention: how does the performer conceptualize the relationship between their physical gestures and the sonic results? This is where issues of consistency, continuity, and coherence come into play: does the instrument produce the same sound given the same gesture? Does a slight change in gesture result in a slight change in sound? And, finally, do the sounds and gestures relate in ways we are habituated to from the physics of the non-synthetic world—such as having a larger gestural force result in louder sound. We refer to this as the cognitive retraining problem, because to solve it performers have to retrain how they conceive of the sounds being produced. The system we describe attempts to minimize the cognitive retraining by allowing the designer the designer to associate particular points in the control parameter space with particular sonic results; the designer can therefore choose a mapping that involves minimal reorientation for the performer. From this association of pairs of points, the system creates a mapping from the space of control parameters to the space of synthesis parameters; this mapping is consistent, continuous, and coherent (at least no less so than the underlying controllers and synthesis algorithms are).
You can build an instrument that is consistent, continuous, and coherent in both gesture and sound space and yet is a poor instrument if it does not allow for sufficiently nuanced sounds: simple mappings from a control value to a synthesis parameter can result in instruments lacking in controllable sonic richness. This is our second problem area. It is also the cheesy synthesizer problem: no matter how well you play it, the sounds get tiring because there is insufficient subtle and controllable variety available. In the case of pitch, for example, as the pitch is changed in an acoustic instrument, the timbre automatically changes in various subtle, coordinated ways due to the physics of strings and resonances. If we simply change the pitch of most simple synthesis algorithms, such as FM, without making any adjustments to timbre, the result sounds insufficiently characteristic, that is, it leaves the resultant acoustic gesture underdefined and therefore sounding too simplistic. Such is the case with using pitch bend on an FM synthesizer. However, even in the case where one parameter is used to drive several, such as when the single parameter of MIDI note-on velocity is mapped to many timbral parameters, it can still be perceived as too simplistic: after extended listening, the timbre is understood to lie along only one continuum.

Note well, it is not necessarily the synthesis algorithm's fault, nor the synthesizer's: it is not FM synthesizers that create bad synthesizer sounds, for example, it is that very few people have, first, taken the time to retrain themselves to play the instrument and, second, taken the time to extensively customize their instrument to take advantage of all the control possibilities to a sufficiently rich and subtle degree, and third, that the instruments themselves do not provide a rich enough control space.

One might then think that all the instrument builder needs to do is supply as many controls into the synthesis as possible. However, this can lead to a cognitive overload problem: an instrument may have so many controllable sonic parameters that performers cannot attend fully to all of them at once: they need a mental model simpler than brute-force awareness of every detail. We desire to simplify the performer's idea of the instrument and the sound it produces without impoverishing the sonic output or overloading the attentive capacity of the performer. The simplification we use allows the performer to deal with functional rather than operational tasks: to focus on what perceptual result to produce, rather than what mechanical actions are required for that result. This desideratum is common in the field of computer-human interface design.

All of the above problems—physical retraining, cognitive retraining, richness of sonic results, and cognitive overload—we address by new techniques for designing control and sound spaces for a given synthesis algorithm and automating the mappings between these spaces based on designer-specifiable criteria. The remainder of this paper presents these techniques.

2.1 Performance Mapping

As a test platform, one of us (Goudeseune) has built an "input device": an electric violin tracked continuously in pitch, amplitude, and full spatial position/orientation of bow and instrument body. In combination with various independently defined parameter mappings and real-time synthesis algorithms, this violin controller then becomes a complete instrument useful for investigating these constraints of the performer. The violin uses VSS [Bargar94] for sound synthesis, fiddle [Puckette98] pitch tracking, and the Ascension SpacePad for motion tracking. This input device enables a trained violinist to feel relatively "at home" with the instrument without extensive physical retraining. However, to solve the cognitive retraining problem we turn to a discussion of our mapping techniques.

We consider the family of sounds produced by a musical instrument as lying in a Euclidean space. The axes of this perceptual Euclidean space are given by parameters such as pitch, loudness, and various psychoacoustic measures of spectral content. A similar space is given by the control parameters of the instrument (which may be both more numerous and more difficult to deal with abstractly). The mapping from this control space to the timbral space can also be analyzed: continuity, monotonicity, hysteresis and other mathematical properties affect the simplicity of the performer's mental model of the instrument.

In the case of synthetic instruments, a third space is given by the parameters which the synthesis algorithm accepts as inputs.

One of the central issues in our work is to find ways to mediate between these different spaces. One mode of this mediation is addressed
by providing supplementary visual cues to the performer to help maintain their orientation and direction in a complex space, a kind of feedback score (Garnett, work in progress). The mediation described in the present paper attacks the problem essentially by automatically mapping a complex control space into a conceptually simpler one.

2.2 Control Spaces and Perceptual Spaces

The effective "size" of a perceptual space is proportional to the variety of sounds it contains and the fineness of possible discrimination between similar sounds. Beyond raw size, perceptual spaces can have different kinds of connectedness, or topology. The simplest topology is a product of intervals (a hyperrectangle). But some parameters require other topologies: vowel timbre, described as a circular "hue" and an "intensity" (distance from schwa); or fast rhythms blurring into low pitches, where rhythm and pitch though perceptually distinct share a common axis. Even when parameters are all linear and independent, certain regions of the hyperrectangle may be excluded in a given composition or style of performing, leading to a topological space with gaps or holes. The simplest example of a gap is the low range of the trombone, where certain pitches between normal playing and pedal tones are unplayable. A more common topology is a convex subset of a hyperrectangle: a product of intervals, with some extreme regions removed. For example, the flute can play the pitch B6; ppppp; or for 30 seconds, but not all three at once.

Parameters may also be effectively cross-coupled, in that the performer naturally thinks of certain parameters as varying together in predefined patterns: on the flute, higher pitches are louder, lower pitches are fuzzier or breathier.

2.3 Kinds of Mappings

In our view, it is rarely the case that direct maps from individual controls to individual synthesis parameters will result in useful individual perceptual parameters that are also reasonably rich sonically. Though using a controller with a frequency control (such as keynumber in a midi synthesizer) to drive the pitch of a synthesis algorithm indeed maps the single control parameter to the single perceptual parameter (pitch), such simplistic maps lead, as we have noted above, to an undesirable lack of richness in the resultant sound.

To avoid such oversimplifications of sonic gesture, we consider the general case: one control may drive several parameters, and one parameter may be driven by several controls. Furthermore, we need a way to define sometimes very complex trajectories within the parameter space. We address these needs with, for the former, a general method of geometric mapping called simplicial interpolation, and, for the latter—the creation of rich trajectories within a given space—by using the notion of a timbre rover which one of us (Goudeseune) has developed. The remainder of this paper will describe these concepts.

2.4 Technical description

Often the number of synthesis parameters is much greater than the number of controls. Perceptually interesting mappings from a low to a high number of dimensions can be done with techniques of high-dimensional interpolation. [Pels95] has done so with extensive training of neural networks; instead of various ad-hoc methods proposed in the past, we prefer the general geometric method of simplicial interpolation, a refinement of the method described in [Choi95]: a pointwise map is extended to a continuous, piecewise linear map on the whole low-dimensional space by (a) triangulating the original set of points in the low-dimensional space to form a simplicial complex, (b) inducing a corresponding simplicial complex in the high-dimensional space, and (c) defining a simplicial mapping between the two complexes, identifying corresponding simplices and identifying points in such simplices with equal barycentric coordinates. To illustrate, controlling three parameters with two: the pointwise map is defined by a small number of points in a volume (corresponding points in the plane can be chosen manually or automatically with a genetic algorithm); a triangular mesh is laid on the points in the plane; this mesh is then mapped onto the corresponding "crinkled" triangular mesh embedded in the three-dimensional volume.

This geometric, numerical mapping from control parameters is still only to synthesis parameters. We want to extend the mapping all the way to perceptual parameters, but prefer to do so without tedious manual crafting of the timbre
space. To this end, we use a technique we call a timbre rover. The timbre rover is given a "black box" synthesizer which takes a set of parameter values as input and produces a sound as output. It searches for a small collection of sounds from this synthesizer which match our specified criteria: they may vary maximally, vary maximally within certain bounds, etc. The idea is that interpolating between these settings (with a high-dimensional interpolator as described above) will cover the specified timbral range with as much continuity as the space allows. Roughly, the timbre rover tries out thousands of parameter settings, "listens" to the frequency content of each resultant sound, and judges how different pairs of sounds are. It takes the Fletcher-Munson corrected amplitude-versus-frequency plot of each sound, divides it into critical bands, and notes the loudness present in each band. To define overall loudness, each band is divided into a general noise floor and zero or more spectral peaks. The noise floor is defined as the median loudness of all frequencies in the band. Spectral peaks are defined as local maxima exceeding the noise floor by more than one average deviation; they consist of a frequency, loudness, and width. The distance between two sounds is then defined as the Minkowski metric (p = 5) of the differences of their critical band loudnesses, a generalization of the distance measure in [Feiten93].

3. Conclusions

The techniques presented are flexible enough to yield a wide variety of resultant control spaces with, we believe, most synthesis algorithms. They give instrument designers a new tool for specifying the relationships between physical gestures and sonic result that can satisfy design criteria as diverse as cognitive simplicity and maximal sonic variety. With careful attention to design they should breathe new life into old algorithms and make it easier and more rewarding for performers to learn new instruments.

4. References


