Relationship-Based Instrument Mapping of Multi-Point Data Streams Using a Trackpad Interface

Kevin Schlei
University of Wisconsin Milwaukee
3223 N. Downer Ave.
Milwaukee WI, 53211
1-(414)-248-0759
kevinschlei@gmail.com

ABSTRACT
Multi-point devices are rapidly becoming a practical interface choice for electronic musicians. Interfaces that generate multiple simultaneous streams of point data present a unique mapping challenge. This paper describes an analysis system for point relationships that acts as a bridge between raw streams of multi-point data and the instruments they control, using a multi-point trackpad to test various configurations. The aim is to provide a practical approach for instrument programmers working with multi-point tools, while highlighting the difference between mapping systems based on point coordinate streams, grid evaluations, or object interaction and mapping systems based on multi-point data relationships.

Keywords
Multi-point, multi-touch interface, instrument mapping, multi-point data analysis, trackpad instrument

1. INTRODUCTION
There have been many approaches to developing useful data streams from single-point XY interfaces beyond the position of the two axes. Multi-point interfaces can also use these approaches, generating the same data polyphonically.

Beyond using multi-point interfaces as polyphonic single-point systems, the implicit qualities of a group of points suggests new possibilities for data streams based upon the relationships between points. This paper seeks to explore the creation and mapping of these relationship-based data streams for intuitive and expansive control over sound production.

The first part of this report describes a selection of XY axis research approaches, followed by an overview of multi-point interface implementations. After this, the motivation behind the relationship analysis approach will be clarified, followed by the touchpad instrument designs and performance evaluations. Finally, directions for further use of the relationship analysis system will be suggested.

2. PREVIOUS AND RELATED WORK
2.1 XY Axis Systems
XY axis interfaces have been part of computer music performance for some time. In 1982, the SYTER project used a graphic tablet to control an arbitrary number of parameters using X and Y position [1]. Each parameter was directly controlled by the coordinate position, with the option to scale the value. Since the 1970s, David Wessel has developed multiple XY mapping strategies for control over timbre manipulation using various models of data interpolation [17][4].

Interpolating systems compound the number of controllable parameters on an XY surface by creating multiple origins for position calculation. The overSYTE granulator included an XY parameter control area where the position of the mouse was compared to the four corners of the rectangular space, allowing for control over four parameters. A successor to that control interface, Metasurface, allowed for an arbitrary number of origin points to be placed anywhere in the XY space, and included a smooth interpolation system for transitioning between parameter areas [6]. These examples show how coordinate data can be viewed as relationships to sets of origins, and form part of the inspiration for the analysis procedures discussed later.

2.2 Other Multi-Point Mapping Solutions
Multi-point touch screen interfaces are often used for onscreen object interaction. In Toshio Iwai's 'Composition on the Table,' graphic objects such as switches, dials, and turntables are used to direct synthesis [9]. The Stereotronic Multi-Synth Orchestra introduces a rotary sequencer design on screen that can be manipulated by multiple participants for collaborative performance [3]. Here the complexity of the instrument is loaded on the object side. The multi-point interactions provide an elegant mode of interaction, and the onscreen objects react in a complex fashion. Davidson and Han's large scale interface provides organization and control of unit generator widgets for parameter manipulation, expanding on similar design principles used in the JazzMutant Lemur [7][10]. These projects provide complex behavior control through the ability to control multiple parameters simultaneously. In addition, each incorporates some parameter objects based on multi-point gestures (such as pinch, rotate and trace).

Another approach to multi-point data mapping is using point data for direct control of synthesis parameters. Christian Bannister's 'Sound Storm' tracks point coordinates to alter filter and synthesizer playback functions [5]. The entire surface space can become the control area, and tracking multiple touches is used for voice adjustments as well as altering sequence ranges through gestures.

Two important predecessors to this paper's research are the reacTable* and Balz Rittmeyer's 'Akustisch' [11][15]. The reacTable* uses physical objects as synthesis unit generators, and the proximity, distance and rotation of the objects is used for parameter control. This approach to finding relationships between points is expanded on by Rittmeyer. 'Akustisch,' a touch surface with onscreen visual feedback, is a system where performance is driven by gestures. Here there are no onscreen objects. Points are used as a means to determine complex gestures, and synthesis is driven by recognizing and mapping those gestures to synthesis parameters. Both the reacTable* and 'Akustisch' focus on multi-point relationships for parameter control. It is this approach to multi-point mapping that is the focus of development for this project.

3. MOTIVATION

Golan Levin warns of ignoring the intrinsic elements of your data when working within the extrinsic confines of a grid or set of axes [13]. This thought helped focus the design process on the extraction of useful data unique to a system with multiple points. Arbitrary grids, onscreen object interaction, and data streams typical of single-point XY interfaces were excluded from the mapping process. The goal was multi-point performance as an instrument, not as a polyphonic implementation of an existing XY interface approach, or as a vehicle to interact with virtual onscreen controllers. Points would be treated generically, rather than as objects with specialized functions like the reacTable*.

To isolate multi-point data as the mode of interaction, the design of the instrument had to use data streams that are dependent on the relationships between points. For this reason, mapping data typical of traditional XY interfaces, such as a point's X and Y coordinates, was intentionally avoided.

The analysis of point relationships became the focus of the research. This added step in the mapping process lead to a tiered approach to programming the instruments, one where analysis acts as the bridge between raw performance data and instrument mapping.

4. IMPLEMENTATION

4.1 Hardware

The Apple Macbook Pro trackpad, measuring 5" diagonally, uses a capacitive grid to detect multiple finger touches [2]. Capacitive grids offer sensitive touch response and high levels of accuracy [18]. The trackpad reports each finger's touch position, timestamp, X and Y velocity, finger blob size and angle of finger ellipsoid.

The size of the trackpad is sufficient for expressive performance, but limits some types of interaction. A larger interface would be easier to divide into sections, for example to separate left and right hand input as with 'Akustisch.' The diminutive size, however, can be advantageous for quick transits across the touch area. The trackpad also lacks the visual feedback of a multi-touch screen, where users can touch an image directly. While this does prevent the ability to easily interact with onscreen objects, a visual feedback display on the laptop screen can be created to guide performance.

4.2 Software - Relationship Analysis

This project's data mapping process was designed to work in four tiers: point data gathering, relationship analysis, distribution, and mapping and implementation (Figure 1).

Relationship analysis is a creative part of the implementation process, where new axes of interaction are created based on the comparison of individual points to one another. Points' coordinates, creation times and motion are collected in snapshots and analyzed. This finds relationships such as each point's distance to other points, angle to other points, velocity as compared to other points, and time added and removed as compared to other points. Each snapshot calculates the total and average distance between points, total and average angle between points, total and average velocity, number of active points, area covered, and average position. The rate at which points are added is averaged over time, as well as the number of points held continuously. Points can be compared to all other active points, a specific sub-set of points, or points that have been removed.

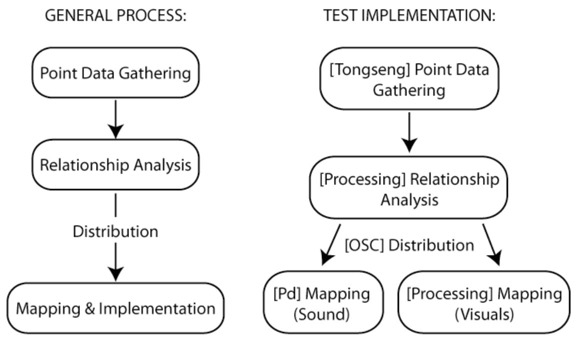

Figure 1. A tiered, four-step approach to implementing raw multi-point data streams

Each new relationship adds an 'axis' of interaction to the multi-point trackpad, making a two dimensional (three dimensional including time) surface a multi-dimensional interface. There are varying degrees of independence between relationships. For example, a group of points moving parallel to one another changes the average position, total velocity and average velocity values without altering the angle and distance relationships between points. In another example, points created in the same positions but in a different order will create different distance and angle relationships but the same final average position. However, some relationships are always linked. For example, a change in distance between points will also result in a change in velocity for at least one point.

It was hoped that mapping a combination of multiple relationships to synthesizer parameters would give the multi-point interface the depth and versatility of a physical instrument, where simple combinations of interaction can lead to complex results.

4.3 Software - Platforms

The open source tools Tongseng [16], Processing [8], and Pd [14] were used to build a four-tiered approach to multi-point mapping. Tongseng is a Cocoa application built to pull the raw multi-point data from the Macbook Pro trackpad and output it via OSC according to the TUIO specification [19][12]. A Processing sketch then takes a snapshot of the point information each frame and analyzes the relationships. Once the analysis is complete, the data is distributed (again via OSC) to Pd, and also used by Processing to draw points and their relationships to screen. Pd maps the analysis data to synthesizer parameters to generate the sound (Figure 1).

4.4 Polyphony
To avoid designating points as unit generator objects, like the reacTable*, each point was allocated its own complete synthesizer voice. The relationships that exist specifically for that point, such as the distance to the point before it, are then linked to control its voice parameters. Global relationship data, such as averages and sums of all point data, are also mapped to individual voice parameters.

5. DESIGN AND PERFORMANCE
5.1 MStrechSynth Design
MStrechSynth is a project implementation which maps angles between points to the synthesis parameter of pitch, distances to amplitude and modulation rate, creation time between points to attack and release time and total speed of points to the delay rate. Inter-point comparisons, such as angle and distance, happen between the 'primary' point (the earliest activated point in any given group) and the subsequent 'satellite' points. This provides a clear relationship for the performer and avoids an overly complex set of distances and angles to control. Performers interact with the instrument by building clusters of points, and then manipulate the sound by changing the relationships between the primary point and satellite points.

Visual feedback is designed to aid performance (Figure 2). The primary point is drawn to screen with a different color for easy identification. Without this information, it would be impossible to determine which point was primary when activating a simultaneous chord of points. Players can follow the angle relationship of satellite points by watching angle arcs drawn around the primary point, which are drawn as filled circle segments. Distances are drawn as lines connecting points. A secondary visual cue comes from the amplitude of each point's voice as reported from Pd back to Processing. Each point grows in size according to the amplitude of the synthesizer voice, so the pulse and loudness of each voice can quickly be associated with its corresponding point.

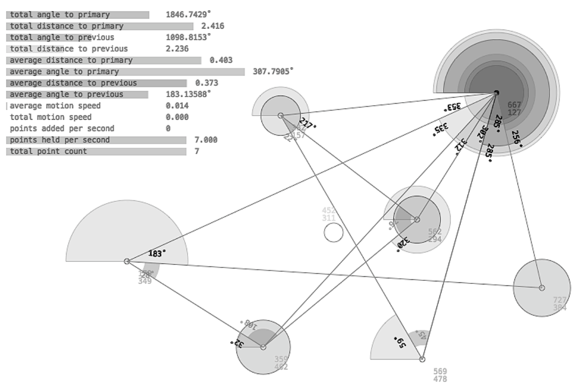

Figure 2. Onscreen visual representation of angles, distances and global relationships between points

5.2 MStrechSynth Performance
Spreading satellite points away from the primary point increases their voice's amplitude, and twisting or rotating satellite points around the primary point changes their pitch. Moving rapidly across the touchpad causes all tones to swell and echo according to the total speed of the points, and is the most visceral performance element.

The instrument was presented to the members of the Milwaukee Laptop Orchestra (MiLO) for evaluation. Performance of MStretchSynth was described as responsive, where the sonic result was easily manipulated with the stretching motion. Expressive gestures could be reliably repeated to produce the same sonic results.

The group agreed that manipulating a small number of points (two or three) is easy to control, but when many points are used it becomes difficult to perform them all independently and still maintain the same degree of precision. Constraints of finger positions can limit the motion possibilities. One suggestion was to divide the performance between human and computer, where one would control the primary point and the other would manipulate the satellites.

The total velocity of points, and its control over delay depth, was considered the most interesting parameter mapping. Performances often center around the 'throwing' of points to create an echoing delay.

5.3 MDrumSynth Design
MDrumSynth takes advantage of the many data types generated when relationship-based analysis is used. The relationships used include angles and distances between sequentially created points, time between point creation, number of points created per second, average number of points held per second, number of active points and average X and Y coordinates of all points (excluding the last).

The angle and distance from each point to its predecessor determine base pitch and brightness of each MDrumSynth voice, where the base frequency is based on the angle, and the brightness increases with distance. Tapping in short steps around the touchpad creates a dull sound, whereas leaping large distances across the touchpad creates bright hits. Rhythmic speed also contributes to the synthesizer's tone. The time measured from one point's creation to the next and the number of points created per second also contribute to sound brightness and resonance. Tapping at an increased speed brightens the tone, allowing for dynamic changes during rolls.

The average X and Y coordinate position filters each MDrumSynth tap, producing sharp or dull responses depending on position. By holding a cluster of points within one area of the touchpad, the average X and Y coordinates will center around that area. That position is compared to the center of the surface, where the farther from the center it is, the more prominent the resulting effect will be on the next voice. The final point is excluded from the calculation of the average, allowing new points to use the averaged data of the others without skewing it towards their own position. The result is similar to the acoustic hand drumming practice of muting different parts of a drum head to achieve different tones. More importantly, it demonstrates a key feature of relationship-based analysis. A specific subset of points was determined to be important, so the relationship was customized accordingly.

5.4 MDrumSynth Performance
Activating multiple points simultaneously results in a bright, sharp drum sound. This is achieved by analyzing the creation time between points, which approaches (or is) zero when

playing chords. MDrumSynth maps this data to the brightness and resonance of each voice. When the player performs a chord, the instrument opens each voice's brightness and resonance to full. In contrast, solo touches sound dampened and dull.

The average number of points held per second is mapped to the release time of each drum voice. When many points are held, new tones ring out. This provides a method for the creation of short and long notes during performance.

Again MiLO evaluated the instrument, finding the core practices of chording and timbre control sufficiently expressive, even though the trackpad lacks the pressure sensitivity typical of percussion controllers. Depth of parameter control was determined to be too subtle (for example, the degree of brightness control when changing the distance between points was not noticeable) and have since been updated.

6. CONCLUSION

Relationship-based analysis allows for new and creative control methods to be born from multi-point data streams. The instruments MStretchSynth and MDrumSynth were designed to provide expressive control using the trackpad. Complexity in performance arises from the interaction of simple relationships and the parameters they control.

After experimenting with the design of MDrumSynth, the data types that offered the most value became apparent. Multi-point parameters that accumulate, average, or are focused on the group rather than the individual were some of the most effective to include. In the case of MStretchSynth, following the total velocity of all points contributed to the most unique performance practice of the instrument. To further refine MStretchSynth, more group relationships could be included to multiply performance possibilities.

The most surprising discovery was the expressiveness of chording in MDrumSynth, which does not implement a dedicated chord detecting algorithm. When the members of MiLO spent time mapping the multi-point relationship parameters from this project to their own synthesizers, they requested a number of new relationship streams, but were most excited about the possibility of creating groups of points for new data types. Groups of points could be based on creation time, proximity or other relationships. MDrumSynth reacts to chords because of the timing relationship between points, but the points are not grouped together to create a new data relationship.

Future experiments aim to expand the application of multi-point relationship analysis. Other interfaces beyond the trackpad, notably video tracking of dancers or body positions, have been suggested as possible avenues for exploration. In each, the hope is to find what relationships are important in performance, and map those to create expressive music synthesis.

7. ACKNOWLEDGMENTS

Special thanks to Chris Burns, Steve Schlei and the members of MiLO for their invaluable input and support.

8. REFERENCES

[1] Allouis, J., and Bernier, J. Y. The SYTER project: Sound processor design and software overview. In *Proceedings of the 1982 International Computer Music Conference (ICMC)*, 232–240, 1982.

[2] Apple, Inc. Apple - MacBook Pro - Technical specifications of the 15-inch. http://www.apple.com/macbookpro/specs.html

[3] Archer, Z., and Lewis, R. Stereotronic Multi-Synth Orchestra. http://www.fashionbuddha.com/

[4] Avizienis, R., Freed, A., Wessel, D., and Wright, M. A. Force Sensitive Multi-touch Array Supporting Multiple 2-D Musical Control Structures. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07)* (New York, NY), 2007.

[5] Bannister, C. Sound storm. http://subcycle.org/

[6] Bencina , R. The Metasurface: Applying Natural Neighbor Interpolation to Two-to-Many Mappings. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME05)* (Vancouver, Canada), pages 101–104, 2005.

[7] Davidson, P. L., and Han, J. Y. Synthesis and Control on Large Scale Multi-Touch Sensing Displays. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME06)* (Paris, France), 2006.

[8] Fry, B., and Reas, C. Processing http://processing.org/

[9] Iwai, T. Composition on a Table. http://www.ntticc.or.jp/Archive/1999/+-/Works/conposition_e.html

[10] JazzMutant. Lemur. http://www.jazzmutant.com/

[11] Jorda, S., and Kaltenbrunner, M., et.al. The Reactable*. In *Proceedings of the 2005 International Computer Music Conference (ICMC)* (Barcelona, Spain), 379-382, 2005.

[12] Kaltenbrunner, M., Bovermann, T., Bencina, R., and Costanza, E. TUIO - A Protocol for Table Based Tangible User Interfaces. *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW)* (Vannes, France), 2005.

[13] Levin, G. Painterly Interfaces for Audiovisual Performance. Master Thesis, Massachusetts Institute of Technology, 2000.

[14] Puckette, M. Pure Data. In *Proceedings of the 1997 International Computer Music Conference.* International Computer Music Association (San Francisco, CA), 224-227, 1997.

[15] Rittmeyer, B. Akustisch. http://akustisch.digitalaspekte.ch/

[16] Tongseng. http://github.com/fajran/tongseng

[17] Wessel, D. L. Perceptually based controls for additive synthesis. In *Proceedings of the 1976 International Computer Music Conference (ICMC)*, 1976.

[18] Westerman, W. Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-point Surface. Ph. D. Thesis, University of Delaware, 1999.

[19] Wright, M. Open Sound Control: an enabling technology for musical networking, *Organised Sound*, 10, 193-200, 2005