

IMPROVISING WITH GRID MUSIC SYSTEMS

Roland Adeney Andrew R. Brown
Queensland University of Technology
Victoria Park Road, Kelvin Grove

ABSTRACT

Grid music systems provide discrete geometric methods for simplified music-making by providing spatialised input to construct patterned music on a 2D matrix layout. While they are conceptually simple, grid systems may be layered to enable complex and satisfying musical results. Grid music systems have been applied to a range of systems from small portable devices up to larger systems. In this paper we will discuss the use of grid music systems in general and present an overview of the *HarmonyGrid* system we have developed as a new interactive performance system. We discuss a range of issues related to the design and use of larger-scale grid-based interactive performance systems such as the *HarmonyGrid*.

1. INTRODUCING GRID MUSIC SYSTEMS

We define Grid Music Systems as musical systems that provide a visual grid or matrix layout on a screen or physical interface as a method for the temporal structure of musical content. Moving around on the grid, like a game board, or pressing buttons of a grid-like array, provides the means to operate on, or 'play' the grid. A system may be a hardware console, software application, or a combination of the two. A typical grid system has a checkerboard-style layout with an active marker or cursor that shows movement on it. At times physical grid interfaces comprise a device covered with a square array of 'buttons' that are activated by being pressed. Typically, grid systems allow the user to perform or program a sequence of steps around the grid that produce cyclic musical patterns. Sound output from electronic grid systems is usually provided by virtual synthesizers and heard through a sound system.

Because grid layouts are so ubiquitous, we need to also describe some grid-related musical interfaces that are outside our current definition. There is a 'music grid' on *Facebook*¹, where one can create a grid of pictures of favourite album covers, listen to samples, and discuss them with others. There are various other 'grids' which are collections or databases of music files, such as the

In-Grid MP3 download site. In addition, a chord chart may be referred to as a harmony grid. Such systems are outside the scope of this article.

2. OVERVIEW OF THE *HARMONYGRID*

We have developed an interactive music system called the *HarmonyGrid*. While "MusicGrid" might have been a more suitable name, given that it allows control over several musical parameters, it was not used because it is already widely used to mean other things. The only remaining direct source of confusion may be Levitt's *HarmonyGrid* program for Macintosh [14]. Levitt's program was a mouse-driven music application using a grid, where the speed of mouse movement over the grid affected note duration and position on the grid selected pitches, and cell steps could be adjustable to cover a variety of interval ranges. However chords need be selected for each (different) chord on each pitch.

Our *HarmonyGrid* uses a simple 4x4 grid of squares which is projected vertically down onto a performance area, so that it forms an area 2 metres square. The user, a musical performer, walks on and around the grid. The user's location on the grid is detected by a webcam overhead. Software developed in the Pure Data environment with the GEM graphic library extension, drives both the grid graphics (which are somewhat animated), and the generated music. Changes to the music can be triggered by the performer's location. The grid operates in five modes, corresponding to the musical parameters of spatial position, volume, rhythm, timbre, and pitch (or harmony). Movements of the performer on the grid sets the current mode parameter to the particular cell activated. Musical output runs continually and is generated in real time and interaction is 'live', so that for example, in 'harmony' mode, arpeggios based on the currently triggered chord continue to sound until a new cell is reached. Musical output is rendered with virtual synthesizers, spatialised to a quadraphonic speaker field, covering the performance area. The performer's role is to both improvise over the grid's musical output with a portable musical instrument, whilst moving around the grid, and operating the controller (described below) when desired. A setup diagram and photo of the

¹ The site is available from one's own Facebook profile via the search box, so a reference web address is not meaningful.

HarmonyGrid system are shown in figures 1 and 11 respectively.

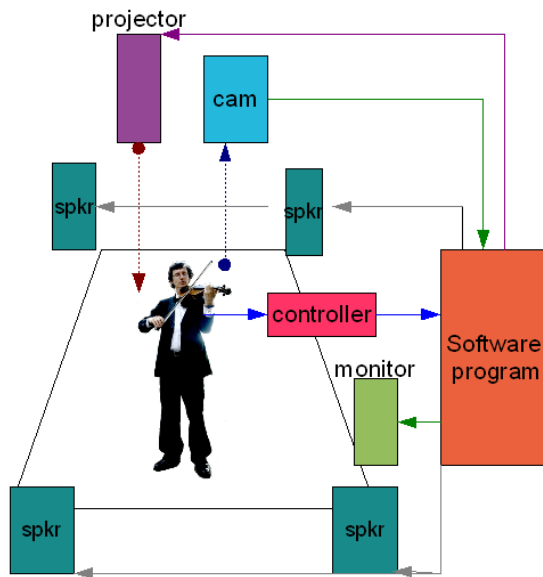


Figure 1. Information flow with the *HarmonyGrid*.

The *HarmonyGrid* is designed as a new musical instrument. It appears to be quite unique, in that it is a (relatively) large-scale grid on which the performer moves around to produce music. The *HarmonyGrid* bears some similarity to a number of previous music and movement systems, but these have often focused on movement as dance rather than movement as simple musical control. Dance games, such as *Dance Dance Revolution*, provide somewhat similar systems, in that they involve moving on a special surface (e.g. the dance-pad). However, the motivation for engaging is dancing and dance accuracy, rather than the creating and controlling of a musical score. Other similar systems where musical control is at least part of the result include interactive installations where sound samples and soundscapes are triggered by a user's location, as well as a few motion-tracking systems for installation or dance performance. Many motion-tracking systems, such as Garth Paine's *Gestation* [9], track full-body motion and allow one or many users to trigger quite complex musical systems. Tracking in Paine's systems is continuous across the space. Discrete dance-based motion tracking was used in *Dis-patch* [18], where quantized inputs – locations of limbs - was used to inform a complex music-making computer program. In these dance systems there is no obvious (visible) geometrical system constraining the input, to inform the output, and as such, is obscure to the observer.

3. REVIEWING GRID MUSIC SYSTEMS

In order to better understand the opportunities and limitations of various approaches to grid music systems we will review a range of different ones. This review is structured around a developing taxonomy of factors relevant to grid music systems. Typically, grid music systems create music from a combination of direct gestural input that triggers predefined music sequences or sound events. Playback often scans through the grid triggering selected cells, and scanning iterates, resulting in a looped output. In a more complex system like the *HarmonyGrid*, multiple synchronous scans or paths through grid space are possible. Few systems provide their own sound output, most requiring additional hardware such as a computer or MIDI instruments, to sound. The *HarmonyGrid* is no exception.

3.1 Topologies

We restrict our review to those systems using square grids as the layout for placement and movement of components, in two dimensions. Naturally, other topologies are possible, such as for the *ReacTogon* [5] that has a hexagonal array of buttons. Structures in many dimensions are also possible, but most systems use 2D grids, as these provide sufficient complexity to manage several parameters in real-time (e.g. time and pitch are common). The grids may be delineated by lines bordering the active spaces, such as a simple grid of squares, or be constructed of a square array of squares, circles, discs, buttons, or locations marked in some way. For example, the *Tenori-On* [12] model shown in figure 2, is a hardware grid providing a 16x16 array of circular buttons.

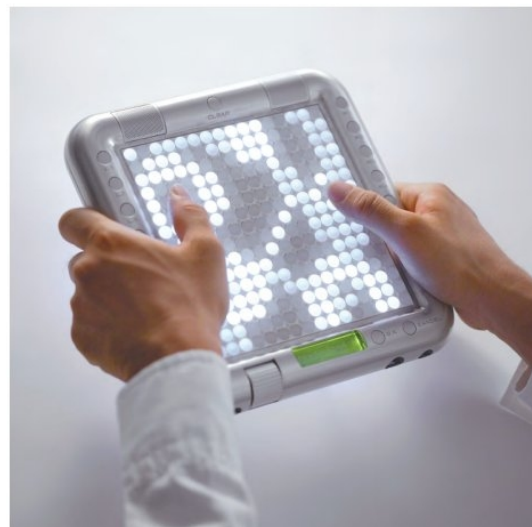


Figure 2. The *Tenori-on*

[Reproduced with permission from
<http://www.global.yamaha.com/tenori-on/index.html>]

The grids are often constrained to a small area, as with portable hardware grids, or expandable to user requirements, as with *Nodal* [16], which provides an expandable 'graph-paper' style grid to write to. Based on a rectangular 2D grid, *Nodal* [16] allows for a user-configurable network structure that provides for construction of paths of significant complexity and various topologies. To some extent the degree of flexibility of *Nodal* [16] moves it beyond the scope of our current definition of grid music systems, because it is difficult to assess whether the musical outcomes can be significantly attributed to the underlying grid.

3.2 Activation

Activation of components on grid 'squares' or cells is achieved in many ways. With hardware devices, buttons may be pushed directly as with the *Monome* [6] and *Tenori-On* [12], or objects on the grid may be recognized by video from beneath, as with the Bubble-Gum Sequencer (Hesse et al. 2007).



Figure 3. The bubblegum sequencer
[Reproduced with permission from <http://www.backin.de/gumball/>]

In software-based music grids, activation often occurs by clicking on a cell or an animated icon 'arriving' at a location. Many systems allow for paths to be set up on the grid, using graphical icons to control path directions as well as musical outputs. This is similar to a network of railway tracks with sets of points or switches, and one or several trains running on them. In these systems, current location is indicated by a flashing disc or the players' avatar/icon, as it moves along the paths. The *Al-Jazari* system [10] provides an example of automated icons moving around a software grid (see figure 4). Usually, several active icons are traversing the paths.

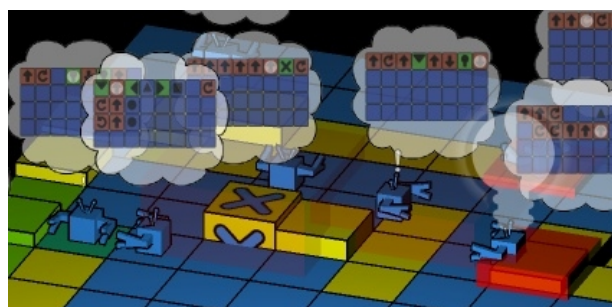


Figure 4. A screenshot from the *Al-Jazari* software
[Image from <http://www.pawfal.org> - Creative Commons License]

3.3 Input

Input for software systems, can be via a mouse, joystick, trackpad, MIDI controller or other device. Input devices typically function to trigger cells or set up and modify paths or sequences with a series of cell clicks. At times they act as tracking mechanisms where the pointer is the active component that is used to 'steer' through grid space. Typically, arrows and other keys on the keyboard may assist in direction controls when step-by-step navigation around the grid is desired. Input devices for hardware grid systems are typically provided by an array of buttons, as with the *Monome* [6] shown in figure 5, or a touchscreen.

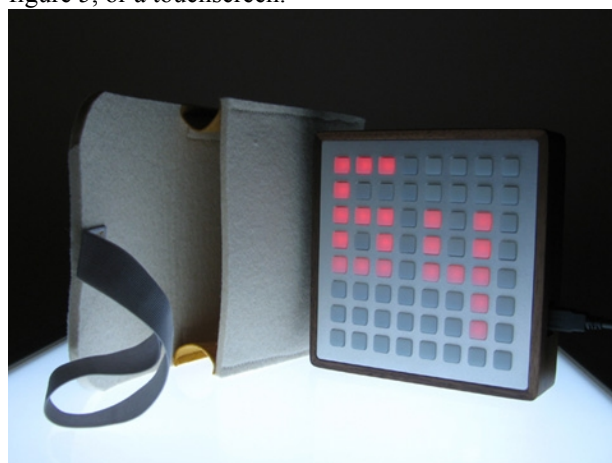


Figure 5. The *Monome* 64 grid interface device
[Reproduced with permission from <http://monome.org/>]

Input can also be via human movement or gesture that ranges from detection of digits or limbs to whole body location in space, or even extend to the detection of geographical position via GPS. The movement input can then be detected directly via hand operated input devices, or remotely by sensing or tracking from a distance. The *Buchla Lightning II* allows two wands to be played or conducted in a 2D grid space in front of the player. Detected by infra-red, the wands act as a MIDI controller to play synthesizers. *Lightning III* extends that control to 3D space.

To display the input, some systems favour a primary active icon, triggered by the user, for example with dance mats as shown in figure 6, whilst many

systems have multiple active icons, only limited by musical intelligibility.

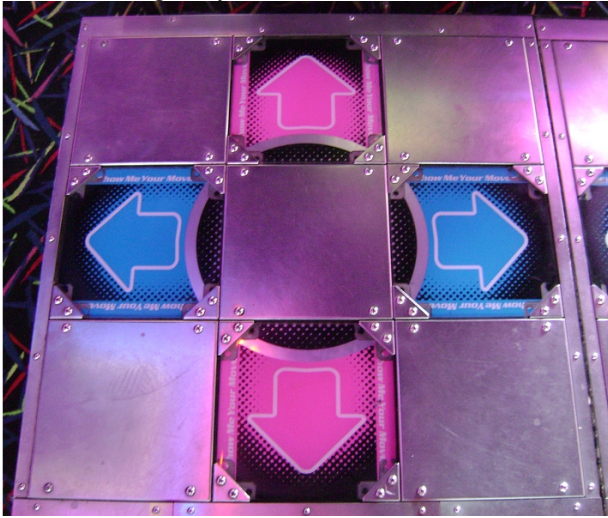


Figure 6. An example of a dance mat grid
[Images adapted from http://en.wikipedia.org/wiki/File:Dance_Dance_Revolution under the GNU Free Documentation License]

Regardless of the input device, system design requires decisions about the mapping from inputs to various parameters, including graphics, music, spatial direction, among others. Mapping processes can range from the direct and simple to the comprehensive and detailed, and discussion of these is well covered elsewhere [8].

3.4 Outputs

Output from a grid music system comprises sound and a visual or graphical display. Some systems may provide haptic feedback. Graphical outputs usually include the grid itself along with the representation of paths, icons showing current location(s), and possibly other system state indicators. Graphical representations tend toward diagrammatic 2D designs, as with Nodal [16], rather than the 3D graphics more typical in computer games, as with Al Jazari [10].

Musical outputs typically are notes, short phrases, arpeggios or samples. These may combine and accumulate to make surprisingly complex resultant music. Tempi are generally adjustable for all systems. Sound palettes for hardware devices may be non-existent, where the device acts only as a controller, like the Monome [6], may be simple, particularly in self-contained systems such as the Tenori-on [12], or may connect to the vast array of software synthesis plug-ins. The audio spatialization capacities of systems will be directly dependent on the capabilities of the software synthesizers and audio hardware employed.

Systems may also send grid location or path position data over various protocols for coordination with auxiliary devices. Typically these protocols include MIDI and Open Sound Control (OSC), but may also include Wi Fi, Bluetooth, and so on.

3.5 Musical Geometries

There are many possible topological mappings of musical parameter space onto a grid layout. Given that a grid has a fixed number of cells - for practical purposes usually (much) less than 100 - choices need to be made about both the selection, quantization or clustering of parameter values and their arrangement on the grid. Topologies may be set up as a simple array of one parameter, such as pitch scales, or cells may relate to each other in two dimensions, as do 'neighbour relations' between the cells in 2D cellular automata. Parameter values may scan horizontally, vertically, diagonally, or in more unusual arrangements such as a spiral pattern. Either way they form a knowledge-space or matrix, relating similar items. Simple musical grids may use discrete pitches on different squares, though informationally this can work better in hexagonal arrays using a harmonic table (accordion style, e.g. *ReacTogon* [5]).

In certain circumstances, grid music systems can operate with arbitrary arrangements, like with a spatially triggered sampler with a matrix of touchpads such as the Akai MPC series, originating in the mid 1980s. Most systems provide for several mappings with different musical geometries for different layouts of squares and their functions and some are highly programmable, such as the *JazzMutant Lemur* [13].



Figure 7. A grid layout on the Lemur

[Reproduced with permission from
http://www.jazzmutant.com/lemur_overview.php]

3.6 Paths

A path describes a series of steps or movements around the grid. Paths are created by moving around on the grid, and may be rendered visible by the system typically by leaving a trail of highlighted cells, as on the KAOSS Pads shown in figure 8. While all movement around the grid creates a path, they become more significant when paths can be defined and reused. Paths can be created by recording user movements, or by notating some path function as a series of rules or

commands. Paths are sometimes said to be 'tracked', for example with *KAOSSPAD*.



Figure 8. Korg KAOSS Pad KP3
[Reproduced with permission from
<http://www.korg.com/Product.aspx?pd=269>]

Our *HarmonyGrid* system, amongst others, provides for a real-time path 'recording', which is then represented as a moving sequence of icons.

Paths may be abstracted as gestures or contours and subject to scaling or other transformations, but this feature is found on few grid music systems. A path may represent a small data set such as a tone-row, that can be 'performed' in different ways, for example to generate output at multiple timescales or different path orders (forward and backwards). Paths can be 'played' as a cannon or segmented by varying multiple start and stop times.

Music grids use the grid to provide spatialised paths that may intersect at specific points, and have similarities with network topologies. The spatialised paths allow for 'clocked' musical and graphical events to occur, over time. The topologies become a network operating over time, a sort of graphical score, that may remain fixed, change or evolve slowly over time, or be very active (and interactive). (There is some breakdown of the network analogy, when we consider that time is generally taken to be 2D, moving forward.) *Nodal* [16] uses the grid dimensions to precisely measure timings of events. For example, a simple square path of 4 nodes (on the corners), may be set up, to 'play' as 4 crotchets in 4-4 time, and subsequent placement of nodes between the others will provide exact subdivision timings. Most other systems provide for rhythm – a sequence of timed events – where the active path runs over both active and inactive squares, thereby triggering or not triggering sounds in a pulsed sequence. Many operate in 'step-sequencer mode' where the x-axis is time; and the y-axis may be pitch, or a range of tracks with sound events; and a pulse runs across the grid activating it column by column. The resultant rhythm is often simply composed of crotchets, quavers and semi-quavers.

By using paths created and recorded on a grid in a variety of ways, a grid music system can produce complex results, and can be very flexible in its methods. Even given the small data set of paths, and the significant constraints of small grids, it is not hard to imagine how these methods of path creation and use can generate sophisticated outcomes.

4. RELATED SYSTEMS

Other systems, including block systems, touchpad and touchscreen systems, and 'light table' systems demonstrate other methods for music-making, and provide further context for grid music systems.

Block systems are somewhat parallel systems to grid systems, having many commonalities around the options and limitations of spatial relations between cells and blocks. Block music systems use custom-made cubes, somewhat like toy building blocks, which may be placed next to each other to make paths, as in dominoes, or they may be stacked vertically as well. Examples include *BlockJam* [17], shown in figure 9, and *Siftables* [15].



Figure 9. The BlockJam

[Reproduced with permission from <http://www.sonycs.co.jp/IL/projects/blockjam/contents.html#interacting>]

In common with grid systems, block systems may effectively use square grid layouts, allow for construction of paths, and typically use several types of blocks to differentiate path directions or musical outputs. These systems may be quite sophisticated, as in *BlockJam* [17], *Percussa Audiocubes* [19] and the *Tangible Sequencer* [3], where blocks communicate with each by physical contact and communicate with a computer, and the software supports various musical or synthesis patterns created by arranging them. Paths are made by the arrangement of blocks and placement of special path-directing blocks. Sound output is via an external system rather than from the blocks themselves. The *MorphTable* [4] uses blocks somewhat differently. Being encoded with graphical designs, the blocks are identified by the camera and software and trigger musical events. They may be freely moved around on the table to affect music-making.

M, the "Intelligent Composing and Performing System" from *Cycling '74* [7] is a music-making program from the mid 1980s. *M* allows interactive composition in that "You can control your music by clicking and dragging the mouse on the computer screen, by "conducting" in a Conducting Grid" [7] which is a 2D grid area on the interface. *M* uses a one dimensional grid system for cycling around pitch, rhythm, dynamics etc independently of one another. The Patterns window,

contain the Patterns, which are collections of pitches, rhythms and dynamics that may be transformed in any way. "A Voice in M is a "path" through the program that begins with a Pattern." [7]

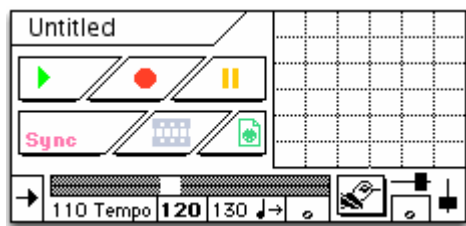


Figure 10. The conducting grid from the M software
[Reproduced with permission from
<http://www.cycling74.com/products/M/>]

Additionally there have been many other 2D surfaces developed to control music, or affect sound, for example AudioMulch's *Metasurface* uses a Voronoi/Delaunay mesh structure [2] in software, to navigate between parameters, sounds or parts of a composition. The geometry may be user-defined.

5. THE HARMONY GRID

The *HarmonyGrid* was designed so that a proficient musician could improvise with and control a generative music system. To that end, the *HarmonyGrid* is a real-time, open-ended format, creating music triggered by the performer's location. Musically, the system is not designed to be sufficient unto itself, but when acoustic improvisation is combined with controller input a satisfying ensemble can be created. Another aim was to provide a spatial component to musical performance that might become part of the theatrical presentation on stage. Spatiality is provided by both the added significance of performer location on stage, and the use of quadraphonic audio projection.

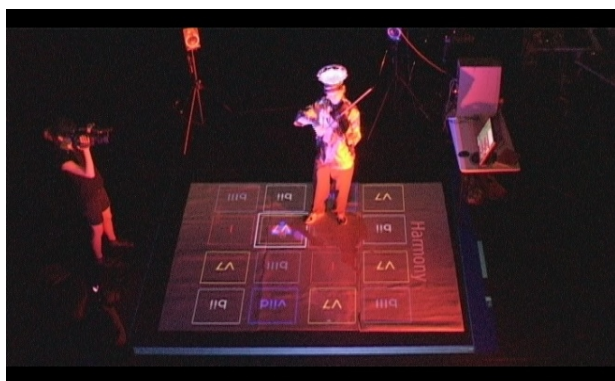


Figure 11. Performance with the *HarmonyGrid* system.

The system can be seen as an extended instrument¹ [1] that augments the instrumental performer, and is designed to provide an interface to electronic/computer music processes that go some way towards the kind of control available on an acoustic instrument.

The *HarmonyGrid* is an immersive grid music system where the performer moves about a large grid projected onto the stage whilst playing an instrument, such as a violin or saxophone. The performer both improvises on their instrument while controlling a generative music system by walking around the projected grid. A camera positioned above the performer, near the projector, tracks the performer's position on the grid. The performer wears an electronic remote controller that can change the settings on the grid and how the position tracking is to be mapped or interpreted.

5.1 Movement on the Grid

The performer may move around on the grid in any way he or she pleases, but the tempo determines when squares are activated. For example, rapidly running around the grid with a slow tempo setting, is not effective. A particular feature is Dance mode, where single pitches are triggered in free time, by 'dancing' around the grid. After performing on the grid awhile, and switching to Dance mode, the main pulse is switched off, yet triggering remains somewhat quantized to maintain some rhythmic integrity. Dance mode now features as a creative interruption, and then the performer can switch back to the previous musical material. Naturally Dance mode adds more of a choreographic quality to the performance, and further consideration of more dance-like movement in the ordinary modes is a subject of future development for the system.

5.2 Music Spaces

The four grids of volume, rhythm, timbre, pitch (or harmony), provide very different musical and graphical environments. Moving on one of these grids alters that musical parameter for the particular square activated.

The program commences with the Volume grid, with its slowly shifting blue discs representative of volume levels, that, combined with the subtly changing arpeggiator, provides a rather static yet gentle ambience. The Rhythm grid, representing rhythmic patterns for each square, tends to be bright and lively, with its capacity for freely changing rhythmic patterns or playing a fixed cycle of rhythms. The Timbre grid is the most colourful, and adds musical timbral effects per square triggered to the musical patterns currently assembled. The Harmony grid presents an intriguing

¹ "“Hyper-instruments” are large scale musical systems which enable the control of complex musical events by a performer.” [1]

display of squares with Roman numeral notation for harmonies, which may facilitate an intellectual approach to selecting harmonic movement. Each grid's musical output can be enriched by adding paths from the other parameters; to good effect.

5.3 Paths and Voices

The performer may choose to record a path from 2-8 steps, which is then stored as the path belonging to the current parameter, e.g., the Volume path. Paths are played, when one or both of the Voices are active. Paths may be written from the current grid to another parameter at any time, e.g. the Volume path may be copied to the Rhythm grid. Or separate paths may be stored for each grid. Paths for each of the other parameters may play whilst on the current grid, so that whilst a harmony pattern plays on the Harmony grid, it may be modulated by the Volume path, the Timbre path (i.e. have differing volume and timbres for each step on the path), and play rhythmic patterns.

The 'active' voice is triggered live by the performer, and has an arpeggiator option. Once a path is recorded, Voice 1 (the upper voice) may be started, with its arpeggiation, and rhythmic options. Voice 2 (the lower voice) (one or both voices may play) runs over the same path, but can be offset by several squares. It has independent phrasing options with rhythms and pitches.

The *HarmonyGrid* currently carries options for harmony grids of various layouts, musical scale selection, choice of 32 instruments per voice, and percussion options. The layout of the harmonies in the harmony grid has been the subject of much research and testing. One layout comes from Euler, and features intervals of sixths running one direction and thirds the other way. Aside from providing a variety of options, the layout does seem to be much less critical than was originally assumed.

5.4 Musical results

As with most grid music systems, *HarmonyGrid* is most suited to loop-based styles, including minimalism, simple groove, and modern dance music. The loops can function as canons, either as chords or single notes, and can scan at very slow to very fast tempi. Very sparse ultra-slow music, to fast frenetic loops are possible. Because loops can function as bass lines, the upper voice can accompany as arpeggiated patterns with subtle changes, providing styles such as simple pop, or richer synth-based dance music to improvise with. Synthesizer voicings provide categories for dance, ethnic, tonal, and more unusual sounds.

5.5 The Physical Controller and visual feedback

The function of the physical controller is to fully enable the performer to move freely around the grid, and not have to return to control the computer via mouse. The controller is really the secondary control, with the

primary control being the spatialised triggering of projected grid squares.

The physical controller is a 14x10x4cm custom-made box hung on the performer's chest. A collection of buttons switches and knobs input to an *arduino* board, which communicates via Bluetooth to the computer. A four layered control display shows on the secondary screen at the side of the performance area, providing visual feedback. The secondary screen also runs the Pd program main page, with its own onscreen mouse-driven controls, as a backup. The controller stops and starts the program, voices, recording paths, playing paths (Voices 1&2), arpeggiators, and rhythm generators; and enables selecting instruments for the voices, and displaying grids. Rotary knobs provide tempo and volume controls. Dance mode and Show-All-paths modes are selectable.

The controller sends rather erratic data that requires smoothing (the non-Bluetooth *arduin*os perform quite stably); with some workarounds in the programming. This early version of the switching layout on the controller, and current version of the control display; make the system somewhat difficult to operate, and more refined versions are anticipated for the future. For instance, familiarity with the switches and display layout is necessary, although all changes are shown onscreen. An important consideration is the necessity to stop playing one's instrument in order to use the controller.

5.6 Experiences of performing with the *HarmonyGrid*

A germinal design idea was for a simple hands-free arpeggiator and pattern accompaniment, and this function is fully realized. With the addition of the multi-parameter path capability, the system has grown into a reasonably sophisticated music generation tool, somewhat unique for its visual display directly mapped onto the performance area. Viewers are stimulated by the overall design concept of direct music triggering, over a graphical environment. Participants have enjoyed themselves, and one small girl danced on the grid for a full ten minutes. Viewers seem to enjoy a walk-through explanation.

Currently the system is still in early development, and quite some practice and experience with it are required to present a smooth performance. As mentioned earlier, the controller and its display screens are somewhat awkward to use, but the paradigm and the overall design takes some digestion and thoughtful consideration on the part of the performer; the instrument needs to be learnt. Spatial determination of volume and timbral paths are simple, yet for harmony, movement control is seemingly a new experience. The whole fascinating subject of spatial representation of music, both mentally, and physically, is a topic for future publication.

5.7 Comparison with other Grid Music Systems

HarmonyGrid is largely different from other grid systems, in the following areas and ways. *HarmonyGrid*:

- uses whole body movement on a projection
- is sonically immersive in a quadraphonic field
- is graphically more engaging than smaller devices with lights or buttons
- requires an improvising musician who understands and can play with written harmonic symbols
- is designed to be an accompanying musical system to live improvisation
- provides an electronic controller
- provides separate 'grids' for musical parameters
- provides layering up of paths from different musical parameters
- doesn't operate in step-sequencer mode
- doesn't provide path direction-switching icons
- features Dance mode, which is 'unlocked'

Similar to most other systems, it provides triggering by squares, and subsequent path activation and recording, leading to the layering up of paths or loops. *HarmonyGrid* doesn't provide output to send to other computer systems such as DAWs, or, for example, act as a controller for *Ableton Live*.

6. CONCLUSIONS

In this paper we have discussed the variety of approaches to grid music systems. We have surveyed a range of systems and discussed salient features of these systems as a class, towards developing a taxonomy of them. Against this contextual background we have described the *HarmonyGrid* system we have developed and provided reflections on its use.

In performance testing, the *HarmonyGrid* has shown to be an engaging and viable means of music-making in a larger-scale grid music system. *HarmonyGrid* demonstrates that spatial control by body location is a viable means to navigate the musical data space. The system is unique in more than several ways; visibly it is unique in that the user walks on a projection of the grid, operates a controller, and in that the system is designed to accompany an acoustic improvising musician. In addition, the system allows layering of paths from various musical parameters.

In future papers we intend to discuss other aspects of grid music systems including compositional considerations, the use of the spatial dimension in traversing musical data, and provide more detailed insights into performance considerations that the *HarmonyGrid* provides as an extended instrument.

While grids as a construct have been used as a musical organizing construct across many cultures and times we hope to have demonstrated that in the digital age they continue to provide opportunities for organizing

and elaborating on music possibilities for the improving musician.

6. REFERENCES

- [1] Anderson, T. "Using Hyper-instruments for the re-distribution of the performance control interface" Reprinted from the *Proceedings of the International Computer Music Conference*, Aarhus, Denmark, 1994.
- [2] Bencina, R. "The Meta-surface – Applying Natural Neighbour Interpolation to Two-to- Many Mapping." *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, 2005.
- [3] Bernstein, J. T. *Tangible Sequencer*. 2007. Retrieved 2 May, 2009, from <http://www.tangiblesequencer.com/>
- [4] Brown, A. R., Wooller, R. and Thomas, K. *The Morph Table: A collaborative interface for musical interaction*. In A. Riddel and A. Thorogood (eds.) *Trans: Boundaries / Permeability / Reification*. Australasian Computer Music Conference. Canberra, Australia: ACMA. 2007.
- [5] Burton, M. *ReacTogon*. 2008. YouTube www.youtube.com/watch?v=AkIKy2NDpqs.
- [6] Crabtree, B. and K. Cain. (2008). "Monome." 2008. Retrieved 2 May, 2009, from <http://monome.org>
- [7] Cycling '74. *M 2.7: The Classic Intelligent Composing and Performing System*. 2007. Retrieved 4 May, 2009, from <http://www.cycling74.com/products/M>
- [8] Doornbusch, P. "A brief survey of mapping in algorithmic composition" *International Computer Music Conference*. Göteborg, ICMA. 2002, pp 205-210.
- [9] Paine, Garth. *Interactivity, where to from here?* Retrieved 4 March, 2009, from <http://www.activatedspace.com/Papers/Papers.html>
- [10] Griffith, D. *Al Jazari*. London, 2008, Pawfal.org.
- [11] Hesse, H., A. McDiarmid, et al. "Bubblegum Sequencer." 2007. <http://courses.ischool.berkeley.edu/i290-13/f07/node/382.html>.
- [12] Iwai, T. *Tenori-on*. Yamaha Music. 2008.
- [13] Largillier, G., Joguey, P. and Oliver, J. *Lemur. Bordeaux: JazzMutant*. 2004.
- [14] Levitt, D. D. and E. Joffe *Harmony Grid*. MIT, Cambridge, MA, 1986.

Adeney, R. and Brown, A. R. 2009. Performing With Grid Music Systems. *Improvise: The Australasian Computer Music Conference*, Brisbane: pp.102-110.

- [15] Merrill, D., Kalanithi, J. and Maes, P. Siftables: "Towards Sensor Network User Interfaces." *1st International Conference on Tangible and Embedded Interaction (TEI'07)*, Baton Rouge, Louisiana, 2007, pp.75-78.
- [16] McIlwain, P., McCormack, J., Lane, A. and Dorin, A. "Composing With Nodal Networks" *Australasian Computer Music Conference*, Adelaide, Australia, 2006, pp.101-107.
- [17] Newton-Dunn, H., Nakano, H. and Gibson, J. *Block Jam*. San Antonio, Texas, SIGGRAPH, 2002, pp.67.
- [18] Parrot, C. and J. Mustard. "Dis-patch/Practice-led Researchers." 2004. Retrieved 10 June, 2009, from <http://practiceledresearch.ci.qut.edu.au/pages/index66b9.html?q=work/shaaron-boughen/dis-patch>
- [19] Schiett, B. "AudioCubes." 2009, Retrieved 2 May, 2009, from <http://www.percussa.com/>.