

Mixer Modeling – An Intuitive Design Tool

Using a hardware controller to actuate parametric design software

David Lemberski¹, Marco Hemmerling²

^{1,2}Hochschule Ostwestfalen-Lippe, University of Applied Sciences; Detmolder Schule für Architektur und Innenarchitektur, Germany

^{1,2}<http://www.hs-owl.de/fb1>

¹david.lemberski@hs-owl.de, ²marco.hemmerling@hs-owl.de

Abstract. *Music and architecture share not only phenomenological similarities in relation to their characteristics - like volume, timbre, tone pitch, instrumentation vs. geometry, materiality, light ambiance or perspective - but imply as well comparability in the process of creation. The investigation of digital tools that cross borders between music and architecture was the starting point for the research project „Mixer Modeling“. Against this background the paper discusses the transformation of a musical composition controller into an intuitive design tool for the generation of architectural geometries. In the same amount that the use of a MIDI-controller increases the degrees of freedom for the simultaneous activation of various parameters the definition of geometric dependencies on the level of visual programming become more important for the resulting geometry.*

Keywords. *Intuitive design tool; parametric design; music and architecture; hardware controller; MIDI; visual programming; human-computer interaction.*

Introduction

Parametric modeling - in contrast to geometric modeling - offers the ability to make fast design changes and thus to explore a plenty design variations in a relatively short time. The resulting shape depends on relationships between modules - which again are base on generative algorithms - and values of the driving parameters. Graphic algorithm editors - like *Generative Components* [1] or *Grasshopper* [2] - establish connections between different modules visually and require no specific knowledge of scripting or programming. The design process is guided by sliders or values, which can be operated gradually. The traditional mouse-screen user-interface however restricts

the designer to change only one or two parameter at a time. This limitation in spontaneity and intuition derives from the constraint given by the two DOF (*Degree Of Freedom*) of the computer-mouse, the x- and y-axis. There are special hardware controllers for CAD available with more than two DOF like the SpaceNavigator [3] from 3DConnexion, a 3D-mouse with six DOF. The sole knob on this mouse allows to control transition and rotation about the x-, y- and z-axis of the model simultaneously or six other parameters respectively. Despite the additional options interconnections between the different parameters and the DOF are still difficult to operate (Lemberski, 2006; Wagner, 2004; Zhai, 1995).



Figure 1
MIDI controller with LED-Rings: Novation ZeRO SL MkII.

Intuitive (Sound) Modeling

The relation between music and architecture has been widely discussed regarding their intellectual and phenomenological levels (Ham, 2005; Martin, 1994; Mueck and Zach, 2007). Parameters and qualities, which are involved in the process of creation and composition as well as processing and modification of music also apply to architectural geometry. Key-aspects in music like tempo, meter, rhythm, dynamics timbre, harmony etc. are comparable to geometric parameters like dimensions, shape/contour (e.g. hard vs. soft), surface properties (texture, structure, color) and spatial features (e.g. positive vs. negative). Furthermore both fields share the same process cycle of creation and modification in the generation of a musical or architectural expression (Riad, 2009; Xenakis, 1971). Against this background the research tries to activate a musical device for the design process in architecture.

Controller devices regarding the creation and processing of music offer an additional peculiarity compared to CAD controller hardware: multiple knobs/faders with only one DOF. With a controller box consisting of 10 faders one can control 10 different parameters simultaneously using 10 fingers.

In contrast to relative controllers like the *SpaceNavigator*, linear potentiometers (faders) and rotary potentiometers (knobs) work in an absolute mode, i.e. their position always reflects their value visually. There exist indeed control interfaces, which use (relative) rotary *incremental encoders* and display its related value through a LED-ring around the knob (Figure 1). There are various music controller interfaces available; they vary in size and number of knobs/faders. Data is being sent to computer through a standard MIDI (Musical Instrument Digital Interface) or USB connection. Installed drivers convert incoming values into MIDI information in case of a USB transmission.

Case study - Mixer Modeling

Setup overview

The hardware controller used in the actual setup is the slim and inexpensive *nanoKONTROL* [4] from Korg (Figure 2). It offers nine faders, nine knobs and 19 buttons. Connected via USB it appears under Windows as a virtual MIDI device ready to control music. *Grasshopper (GH)* is being chosen as software

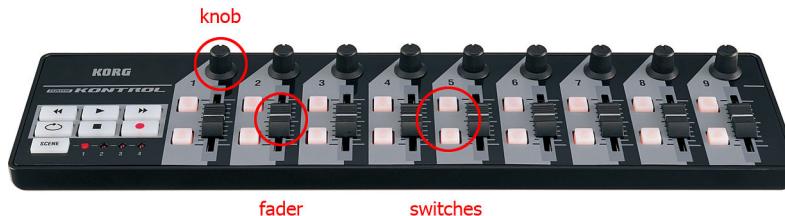
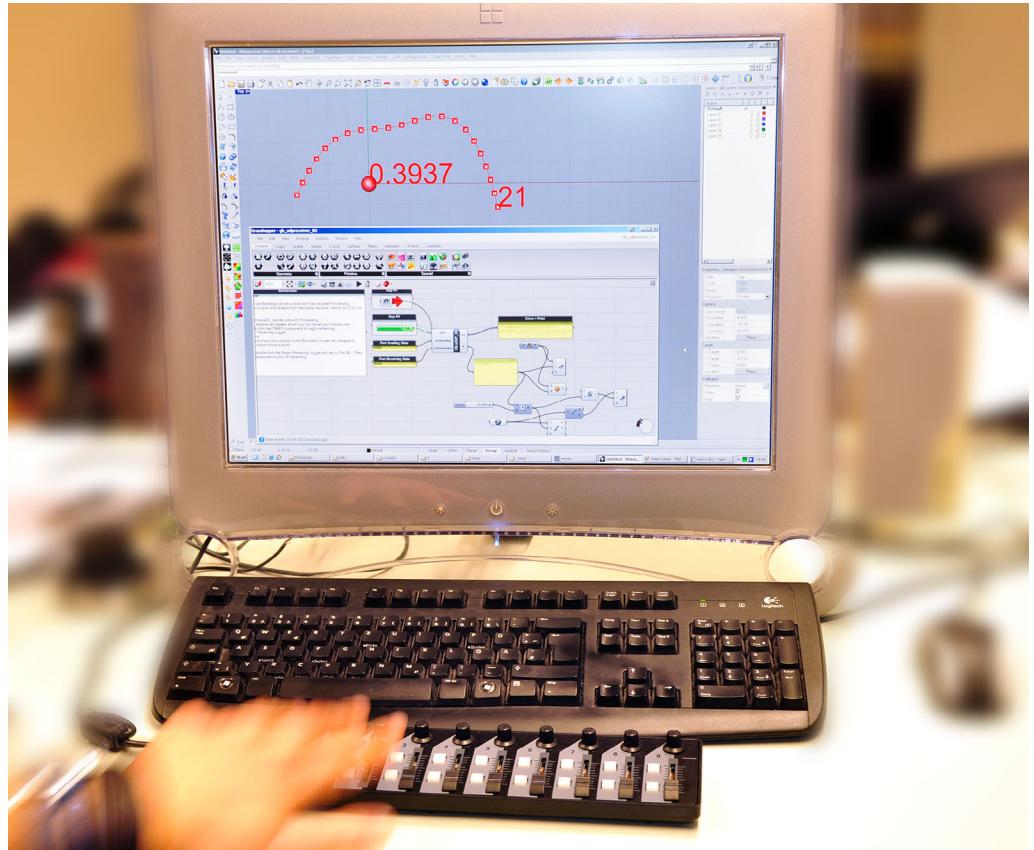


Figure 2
Korg nanoKontrol explanation.

Figure 3
Setup overview.



platform for building and driving the 3D-Model. The set-up runs inside *Rhino* [5] as a plug-in – the geometry can be generated directly in *GH* using varying modules (algorithms) or can be referenced from the *Rhino* workspace (Figure 3). Geometry generated by *GH* appears in the *Rhino* viewport/workspace but cannot be selected or modified until it has been “baked” by the user. Although *GH* offers theoretically direct access to MIDI through scripted components it proved difficult to get it working that way. After some research an indirect way using *VVV* [6] (toolkit for real-time video synthesis) could be found. *VVV* is a multipurpose

toolkit, which in this case has been set up to convert MIDI data and sends it via UDP (User Datagram Protocol) to other programs (Figure 4). A scripted UDP-receiver component written by *LaN* [7] reads the UDP-stream and exposes the contained data to other *GH* modules. Once inside *GH* the incoming values can be connected to other modules and change the assigned parameters (Figure 5). One incoming value can control multiple parameters of the same model through driving different modules working with different algorithms. Dependent on logical complexity (amount of modules, chosen algorithms, referenced geometry) and given update frequency

Now one has access to eight different nodes and values in *GH* according to the eight hardware faders and their positions. The MIDI data range per fader amounts to 127 integer values (steps). *VVV* maps this to a gradual range between 0.0 and 1.0. That range can be adjusted to variable individual needs in *GH* through a special component. Value remapping incorporates shifting and augmentation or diminution of the numerical range (Figure 6).

To keep data flow efficient, *nanoKontrol* sends controller data only while a knob or fader is being moved. Furthermore, always the actual absolute value is being transmitted which makes it easier to access in *GH*. The receiver in *GH* should be switched off when not needed or prior to removing the hardware controller to prevent listening and searching for incoming data and a possible crash of the application.

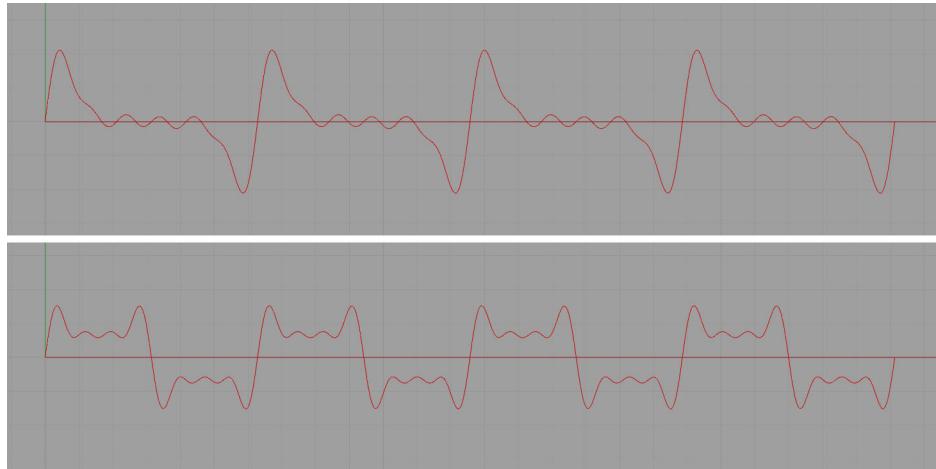
Conclusion

The case study puts in play two different aspects regarding intuitive parametric modeling using MIDI. One part relates to the possibility of a more complex and spontaneous interaction on the hardware level. The other aspect depends on the users ability to wire the modules together, i.e. to make the

logic connections on the software level in order to use the full power and capability of this configuration. Using the MIDI-device one can control miscellaneous parameters simultaneously with the corresponding amount of knobs. The main goal within an architectural design process is to set up mappings, constraints, conjunctions and interdependence between different parameters and algorithms. A combination of multiple single or interdependent constraints is a natural consequence of tasks especially in urban development strategies - as shown for example in the *Kaisersot* [8] project guided by Ludger Hovestadt at the ETH Zürich. Various parameters could be interlinked skillfully together with constraints and thus be controlled by only one or a handful of knobs. Every task requires its individual network of connections and passes through the iterative cycle of creation and modification during the process of development. Subgroups of the defining network however can be extracted to a library and be reused in future projects.

Along with the applied strategies and algorithms the success of the procedure depends mainly on the amount of involved modules and the available processor power how fast the system reacts to changes. Update time can vary from milliseconds

Figure 6
Different fader positions resulting different shapes.



(real-time) up to seconds (slow feedback). The next step after getting a working solution is to optimize it for speed alike known from scripting or programming (Piekarski, 2004).

Outlook

Next steps involve the direct access from inside *GH* to MIDI via a scripted module – abandoning *VVV* from the actual setup - would implicate faster data flow and more effective computation. Other kinds of hardware controllers like single-touch 2D pads or multi-touch pads/displays could be connected to study their influence on the workflow and intuition. Graphical feedback multi-touch devices like Apple's iPhone/iPad would allow to create a custom visual interface with a free selectable alignment of knobs, faders and switches (i.e. order and horizontal/vertical orientation). In that way the appearance of the interface could be adapted to different tasks and by visual grouping represent more logically consistent the influence and hierarchy of the driving values. Furthermore the actual uni-directional dataflow from the hardware-controller to *GH* could be extended to a bi-directional one so that *GH* would send data back to the controller device for visual feedback. For example, important values from within *GH* could be displayed on the iPhone/iPad next to the faders to give the user direct information.

A future research will focus on the development of a continuously morphing and interpolation between different shapes and geometries or sets of values.

References

Fritz, R., Hsiao, C.P. and Johnson, B. 2009, 'Gizmo & WiView: Tangible User Interfaces Enabling Architectural Presentations', *Proceedings of the 29th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, Chicago (Illinois, USA), 22-25 October, 2009, pp. 278-280.

Ham, J., 2005, 'Music and Architecture: from Digital

Composition to Physical Artifact', *Proceedings of the 23rd eCAADe Conference*, Lisbon (Portugal), 21-24 September 2005, pp. 139-146.

Lemberski, D. 2006, *Mensch-Maschine Interaktion, Entwerfen und Präsentieren im 3D-Raum*, Diploma thesis, Hochschule Ostwestfalen Lippe - University of Applied Sciences, Detmolder Schule für Architektur und Innenarchitektur, Detmold.

Martin, E. (ed.) 1994, *Architecture as a Translation of Music*, Pamphlet Architecture 16, Princeton Architectural Press.

Mueck, M.W. and Zach, M.S. 2007, *Essays on the Interaction of Music and Architecture*, Culicidae Architectural Press.

Piekarski, W. 2004, *Interactive 3D modelling in outdoor augmented reality Worlds*, PhD thesis, The University of South Australia.

Riad, M. 2009, 'Musical Deconstruction / Reconstruction: Visualizing architectonic spaces through music', *Proceedings of the 4th International Conference of the Arab Society for Computer Aided Architectural Design (ASCAAD 2009)*, Manama (Kingdom of Bahrain), 11-12 May 2009, pp. 225-233.

Wagner, M. 2004, *Tracking with Multiple Sensors*, PhD thesis, Institut für Informatik der Technischen Universität München.

Xenakis, I. 1971, *Musique Architecture*, Tournai: Casterman.

Zhai, S. 1995, *Human Performance in Six Degree of Freedom Input Control*, PhD thesis, University of Toronto.

[1] www.bentley.com/en-US/Products/GenerativeComponents/

[2] www.grasshopper3d.com.

[3] www.3dconnexion.com/products/spacenavigator.html.

[4] <http://korg.com/Product.aspx?pd=415>.

[5] www.rhino3d.com.

[6] www.vvvv.org.

[7] www.livearchitecture.net/archives/4096.

[8] www.kaisersrot.com.