Toolkit for building your own instrument with Pd

Hans-Christoph Steiner

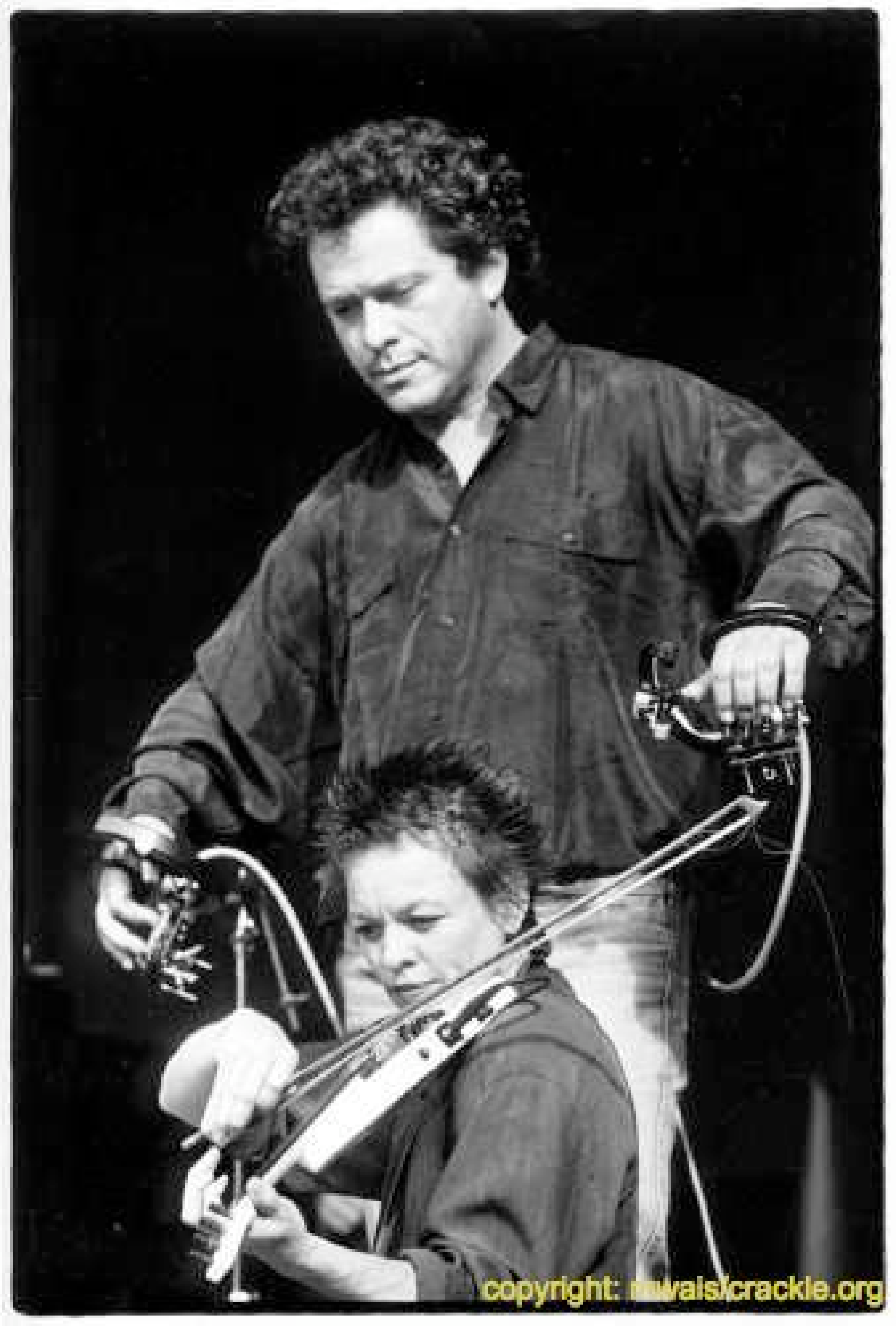
Background

- computers provide incredible, real-time control over sound
- live computer music has become common
- keyboard/mouse/monitor interaction leaves a lot to be desired
- sound synthesis frees the interface from generating sound
- the human body is capable of a massive range of gestures

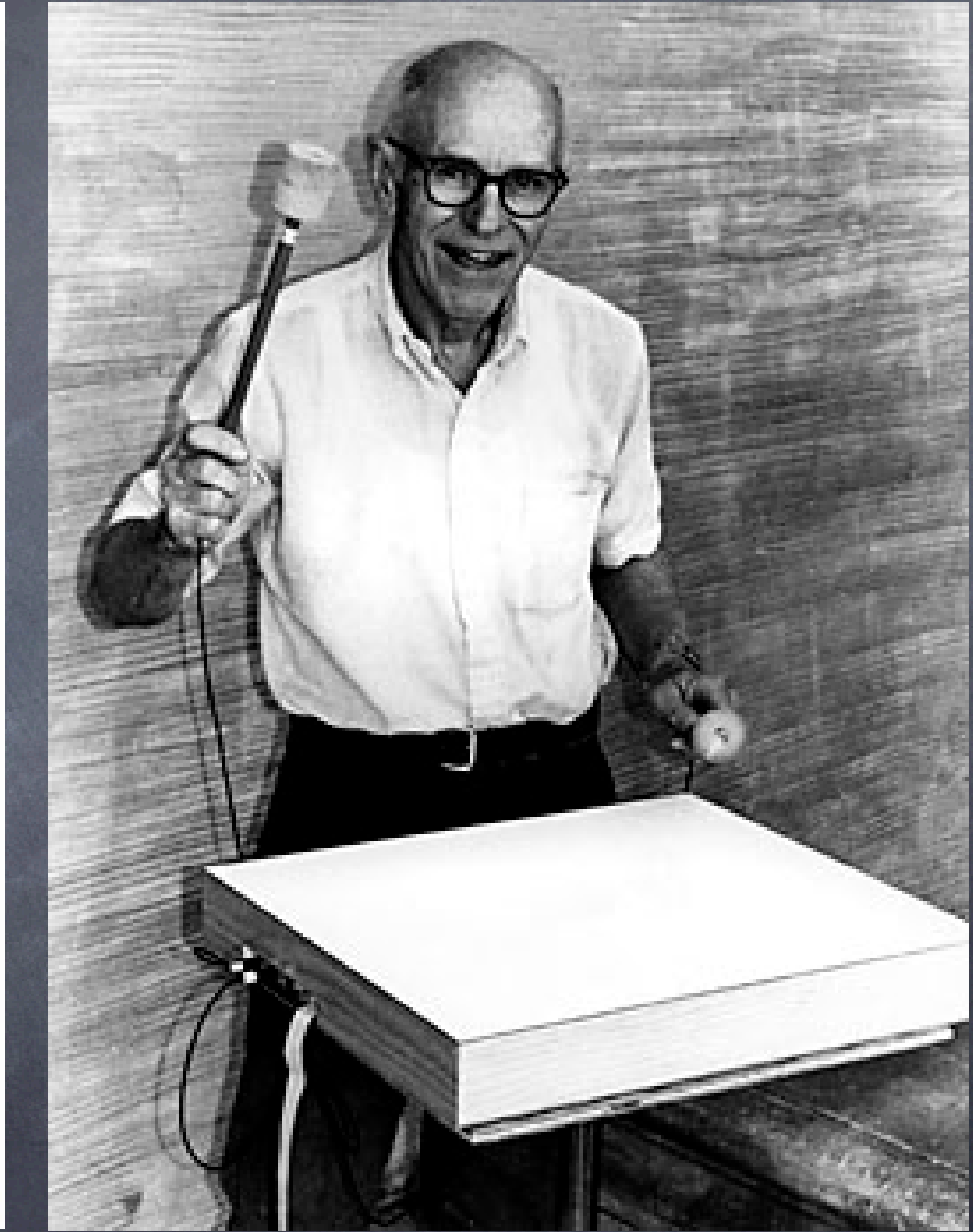
New model of instrument design

- old model: instruments are designed for general purpose and multiple people
- new model: instruments are designed for specific purposes and people
- New Interfaces for Musical Expression

Two examples of new interfaces for musical expression from the mid 80's



Michel Waisvisz playing The Hands



Max Mathews playing the Radio Drum

Three Key Areas

Input

Getting data from the user which is used for control.

Output

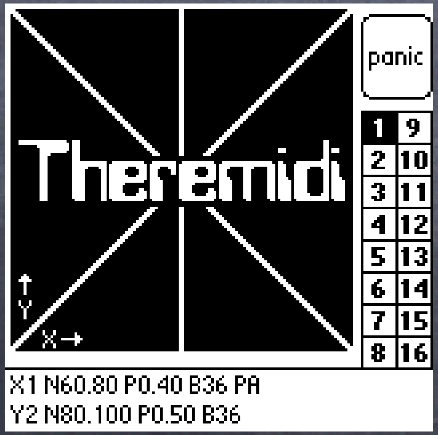
What the user wants to control and create. Traditionally, this has been sound, but it need not be limited to that.

Mapping

Mapping how the input data is mapped to the parts of the program that create the output.

Human Interface Devices (HIDs)

- mice, trackpads, trackballs
- joysticks
- gamepads
- tablets
- gloves
- sensors via microcontrollers or sensor boxes
- video tracking
- custom electronics



X1 N60.80 P0.40 B36 PA
Y2 N80.100 P0.50 B36

Various MIDI devices

Various USB devices

Haptic Feedback

Haptic

Haptic refers to the “sense of touch at the skin level and force feedback information to the muscles and joints”

- joysticks, gamepads, steering wheels, mice
- vibration and forces

Pd provides a fertile environment

- rapid programming environment
- high level programming
- unified platform for audio, video, physical modelling, input devices, haptics, etc.
- free software running on multiple platforms

Pd objects

- [comport] for serial devices and microcontrollers
- [linuxmouse] and [linuxevent] for HIDs via the Linux event system
- [gemmouse], [gemtablet], etc. for HID data from the Gem window
- [joystick] for USB HIDs on Windows and GNU/Linux
- kaos tools abstrations for the Kaoss Pad
- ff library and [ifeel] object
- MIDI objects

StickMusic

Human Interface Toolkit

Currently, Pd and similar environments like Max/MSP or SuperCollider provide much of the ability to work with HIDs. My goal with this toolkit is to provide a unified method of using a wide range of HIDs, while also providing an environment for rapid prototyping of idea for both beginners and advanced users.

- low level [hid] object provides common method to access most HIDs
- high level [mouse], [tablet] objects provide easy access to common HIDs
- high level objects to control haptic feedback effects
- a collection basic mapping objects for rapid prototypes