

Quadrant: A Multichannel, Time-of-Flight Based Hand Tracking Interface for Computer Music

Chris Chronopoulos
Independent Researcher
288 Norfolk St
Cambridge, Massachusetts
chronopoulos.chris@gmail.com

ABSTRACT

Quadrant is a new human-computer interface based on an array of distance sensors. The hardware consists of 4 time-of-flight sensors and is specifically designed to detect the position, velocity, and orientation of the user's hand in free space. Signal processing techniques are used to recognize gestures and other events, which we map to a variety of musical parameters to demonstrate possible applications. We have developed Quadrant as an open-hardware circuit board which acts as a USB controller to a host computer. The software and hardware designs are released under a Creative Commons license.

Author Keywords

NIME, proceedings, L^AT_EX, template

CCS Concepts

•Hardware → Sensors and actuators; Tactile and hand-based interfaces; PCB design and layout;

1. INTRODUCTION

While the majority of musical interfaces leverage the dexterity of the hands for their exceptional information capacity [17], a relatively smaller branch has sought to capture the expressivity of the hands in free-space. Free-space interfaces extend the interaction to 3 dimensions, and by some metrics, hand gestures can yield a much higher level of reproducible information throughput than conventional interfaces [16]. Over the past century, this branch of interfaces has evolved rapidly alongside developments in hand-tracking technology.

Perhaps the earliest example of free-space hand-tracking in music is the theremin, invented in 1920, which uses capacitive circuitry to sense the distance of the hands from two antennas. A unique example of a pre-war electronic instrument, the theremin's high degree of sensitivity gave way to a lineage of virtuosos, and its exposure through art music, film scores, and television contributed to its overall success throughout the 20th century [9].

Starting in the 1980's, musical hand tracking began to benefit from the emergence of technologies which use wearable components to resolve the articulations of the fingers and wrist, in addition to overall hand position. So-called

“datagloves” use flex sensors, accelerometers, and contact points to detect hand shape, and have been used to control virtual instruments and other performance parameters [19], [13]. Another approach measures muscle contractions in the forearm to indirectly detect articulation of the hand [15], [12], [10]. While offering high-resolution detection of hand shape, these interfaces require the user to don unnatural equipment that can hinder free expression.

A more recent class of hand-tracking interfaces uses computer vision techniques to determine hand and/or finger position from video data. Devices like the Microsoft Kinect use a traditional RGB camera in conjunction with a depth sensor to form a 3D model of the scene [27], [18]. Similarly, the Leap Motion peripheral uses a *pair* of cameras to stereoscopically track the positions of each finger individually [11], [24]. Both of these approaches assume models of the subject, and require significant computation to process the data in real time.

A number of gestural interfaces have been developed based on the reflection of wave energy off of the hands. The intensity, phase shift (time-of-flight), doppler shift, or other properties of the reflected waves can be used to determine the distance, shape, and motion of the hands. For instance, infrared distance sensors have been used for musical control, with commercial examples including the Roland “D-Beam” [6], and the Alesis AirFX [26]. These sensors are typically 1-dimensional – they measure hand distance along a single axis – and their sensitivity varies with ambient lighting conditions [23]. Ultrasound is another form of reflective sensing which has been used to control musical parameters to [14], [5]. And more recently, millimeter-wave radar has appeared as a musical interface, offering unique opportunities for motion and gesture control [4].

2. INTERFACE DESIGN

Here we present a human-computer interface (HCI) which performs free-space, unencumbered hand-tracking using an array of distance sensors. The interface consists of hardware for measuring distance and orientation of the hands, and software for processing the data into control parameters. This is implemented on a custom circuit board which acts as a USB peripheral to a host computer. From there, the control parameters can be mapped to any number of musical applications, which we explore in Section 3.

2.1 Distance Sensors

To measure distance, we use four VL6180X sensors from STMicroelectronics, which feature time-of-flight circuitry in an integrated package [21]. Time-of-flight (TOF) is a relatively new distance sensing technology which measures the time taken for an emitted pulse of light to reflect off a target and arrive back at the sensor. This provides an absolute distance measurement which is independent of the target's



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

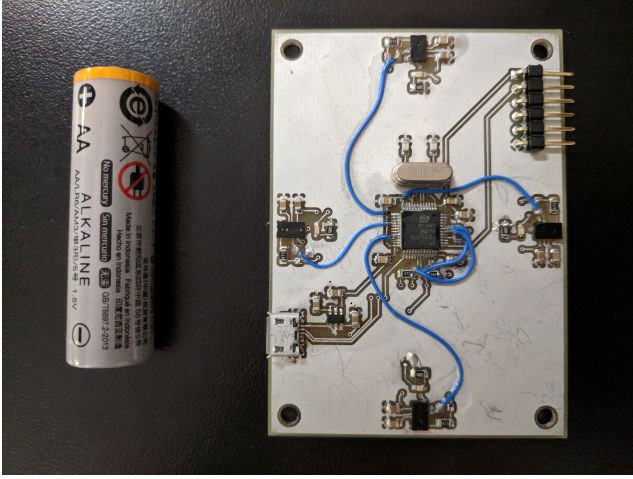


Figure 1: Quadrant prototype PCB (Rev 3), showing considerable re-work. AA battery for scale.

reflectance and the ambient light conditions (within limits). The sensors operate at an infrared wavelength of 850 nm, and have a specified distance range from 0 to 10 cm, though we have observed accurate ranging up to 20 cm or more in favorable lighting conditions. The cone of acceptance for the reflected illumination is 25 degrees (full width), so care must be taken when placing multiple sensors in an array to avoid cross-talk. The sensors communicate over an I2C interface, which facilitates the setting of configuration registers as well as the readout of sensor data. In addition to ranging, they feature an ambient light sensor with sensitivity to visible light (450 nm to 700 nm).

2.2 Circuit Board

The sensors are assembled in a “diamond” formation on a custom-designed printed circuit board (PCB), which is shown in Figure 1. The PCB measures 73 mm by 53 mm; these dimensions were chosen to give the sensors enough spacing to avoid cross-talk, while also spanning the approximate length and width of the adult human hand. Each sensor is accompanied by a blue LED (470 nm to avoid interference with the TOF light) which can be used to indicate status information to the user. For example, we have found it very useful to know which sensors are receiving signal from a hand placed over the array; lighting an LED when a threshold range is crossed provides clear feedback to this end.

The I2C, USB and GPIO are managed by an on-board microcontroller: the STM32F070CB [22]. This chip was chosen for its performance (48 MHz ARM Cortex-M0 with 16 kB SRAM and 128 kB Flash) and for its inclusion of a USB 2.0 peripheral in a low-cost package. The circuit board also features a USB Micro-B connector for power and communication with the host, a 6-pin serial-wire debug (SWD) header for programming and debugging, and a 2-level linear regulator which provides both 3.3 volts for the microcontroller, and 2.8 volts for the distance sensors.

Quadrant adheres to the Open Source Hardware (OSHW) Statement of Principles 1.0 [3], and is in the process of becoming certified by the Open Source Hardware Association (OSHW). The PCB was designed in KiCad, an open-source software suite for electronic design automation. The project files, schematic, Gerber files and bill of materials (BOM) are available on GitHub for anyone to access [1].

2.3 Firmware

The firmware for the Quadrant microcontroller (MCU) is open source and available on GitHub [2]. Here we describe its basic operation at the time of writing, but we expect its feature set to grow, and even branch into multiple versions as the project evolves.

Upon startup, the firmware initializes the required peripherals: GPIO, USB, and I2C. The USB peripheral is set to “device” mode and registers as a Communications Device Class (CDC) — commonly referred to as “USB Serial”. The I2C clock is set to 400 kHz, which is the maximum speed allowed by the sensors.

Upon reset, the sensors each default to I2C address 0x29. In order to address them individually, we need to re-assign their addresses — but this can only be done with an I2C register write. To overcome this bootstrapping problem, the “chip enable” line (pin 4) on each sensor is routed to a GPIO pin on the MCU. Thus initialization involves enabling one sensor at a time, assigning it a unique address and configuring its settings before moving on to the next.

The sensors are configured according to the recommended settings in ST Application Note AN4545, Section 9 [20]. The maximum convergence time is set to 20 ms to allow for a board sample rate of 30 Hz. This sample rate is achieved by “pipelining” the range measurement commands with the readouts: all four sensors are commanded to start range measurements first, and then each result is read out in series.

As each range value is measured, it is compared with a threshold value (default is currently 180 mm), and if the target distance is less than the threshold, the corresponding LED is illuminated. The LED’s are also used to indicate an error condition: they blink in a cyclic pattern if exceptions arise during initialization or I2C transactions.

Once all four sensors are measured, the resulting values (each one byte indicating distance in millimeters) are packed into a buffer and sent over USB serial. If there is no target in a sensor’s field of view, then that channel reports 255, the maximum distance value.

3. GESTURES AND MAPPINGS

In its raw form, the data which Quadrant sends to the host computer is a 4-channel stream of distance values — one for each sensor — updated 30 times per second. For musical control, these channels can be mapped directly to software parameters using a variety of signaling protocols (MIDI, OSC, FUDI [7] etc.) and we expect many users will apply Quadrant in this fashion: as a USB controller with 4 virtual “knobs”.

However, as an array of distance sensors targeting the hand, Quadrant was designed to capture higher-level gestures, and by combining and analyzing the data in various ways, we can leverage this aspect for some unique HCI applications. Here we introduce *aggregates* (values constructed from the combination of Quadrant channels over time) and *gestures* (singular events) which we have found to be useful for musical applications. Currently, these aggregates and gestures are calculated by software running on the host computer; in the future, we plan to integrate their calculation into the firmware.

3.1 Altitude, Pitch and Roll

If we consider that the channels represent the values of a scalar field $d(x, y)$ on a unit circle in the plane of the board:

$$(x, y) \in \{(0, 1), (1, 0), (0, -1), (-1, 0)\} \quad (1)$$

then we can calculate the multipole expansion of that

field. The monopole moment is simply the average distance value:

$$z = \frac{1}{4} \sum_i d_i \quad (2)$$

which we associate with the *altitude* of the target. The dipole moments are:

$$\theta = \frac{1}{2}(d(0, 1) - d(0, -1)) \quad (3)$$

$$\phi = \frac{1}{2}(d(1, 0) - d(-1, 0)) \quad (4)$$

which we associate the *pitch* and *roll*, respectively, of the target. Pitch represents flexion/extension of the wrist, while roll encodes supination/pronation. These aggregates are useful in the implementation of, for example, a theremin whose timbre is modulated by pitch and roll of the hand (see Supplementary Video 1).

3.2 Velocity and Acceleration

In addition to the momentary distance values, we can compute their rate of change, thus measuring the velocity of the target. Given that the board samples are measured at regular intervals $\Delta t = 33.3$ ms, this amounts to a simple delta (first-order backward finite difference) between successive samples:

$$v_i = \frac{d_i - d_{i-1}}{\Delta t} \quad (5)$$

This can be applied either to the raw sensor data, or to the aggregates defined in Section 3.1, effectively adding *rotation rate* as another aggregate.

Similarly, the acceleration can be defined as the second-order backward difference:

$$a_i = \frac{d_i - 2d_{i-1} + d_{i-2}}{\Delta t^2} \quad (6)$$

These derivatives are useful, for example, in mappings which mimic percussion instruments: a positive threshold crossing of linear acceleration can be set to trigger a drum hit, and the downward velocity prior to the trigger can be mapped to the amplitude of the hit.

3.3 Plucking

If a target (e.g. a finger) enters the field of view of one of the sensors from the side, the distance reading will suddenly jump from 255 (no target found) to a smaller value. These discontinuities can be detected by tracking the *engagement* state of each sensor. For example, in Python:

```
while True:
    d = readSensors() # a list of 4 ints
    for i in range(4):
        if (not engaged[i]) and (d[i] < THR):
            engaged[i] = True
            print 'pluck: %d %d', i, d[i]
        if (engaged[i]) and (d[i] >= THR):
            engaged[i] = False
```

where THR is a threshold value; we use 180 mm for consistency with the LED threshold described in Section 2.3.

We call this gesture *plucking* because it resembles the playing of a harp, where each sensor virtualizes an individual string. The resulting effect is similar to that of a laser harp [8], but with a distinct advantage: the height at which the “beam” is broken can be measured and used in the mapping. For example, in implementing a laser harp, we may

choose to map the pluck height to the timbre of the sound, in a way that mimics how the sound of a string instrument is affected by where along the strings it is plucked [25]. For a demonstration of this, see Supplementary Video 2.

3.4 Swiping

Another simple gesture we can define is *swiping*, which is when the hand starts outside of the Quadrant field of view (FOV), then enters the FOV of at least two sensors (one at a time) from the side, and then continues outside the FOV in the same direction. Two sensors are required in order to determine the swipe direction, and it is straightforward to implement a processing pipeline which can detect left swipes, right swipes, up swipes, and down swipes independently.

The algorithm builds off of the notion of engagement introduced in Section 3.3, and extends it with a *priming* variable. For a single dimension, say left-to-right, the priming (P) stores whether the order of engagement transitions has been leading up to a right swipe ($P = 1$), a left swipe ($P = -1$), or neither ($P = 0$). This algorithm is illustrated in a state diagram in Figure 2; an analogous second state machine is used to capture up and down swipes.

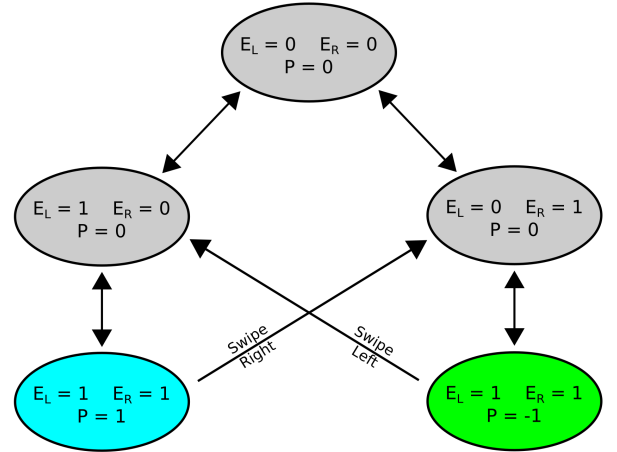


Figure 2: State diagram for detecting left and right swipes from engagement with two sensors (E_L and E_R) and a priming variable P .

One use for swiping in computer music is as a mode selector. A user can have several presets prepared in a piece of software (e.g. a synthesizer), and then swipe left/right to navigate the various settings. For a demonstration of this idea, see Supplementary Video 2.

4. CONCLUSIONS

Quadrant integrates four time-of-flight sensors into a palm-sized, open-source circuit board designed for hand tracking in free-space. We have found that the interface is capable of accurately measuring distance up to 20 cm along four lines of sight at a “frame rate” of 30 Hz. We have defined aggregate measurements related to altitude, pitch and roll, first- and second-order derivatives, and two types of gestures that we have found useful for musical applications. Owing to the open nature of this project, we expect that a many more applications will emerge as the interface is shared with the computer music community.

5. REFERENCES

- [1] Chris Chronopoulos. quadrant-board on github, 2019.

- [2] Chris Chronopoulos. quadrant-firmware on github, 2019.
- [3] O. S. H. Association. Open source hardware (oshw) statement of principles 1.0, 2019. [Online; accessed 27-January-2019].
- [4] F. Bernardo, N. Arner, and P. Batchelor. O soli mio: Exploring millimeter wave radar for musical interaction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 283–286, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.
- [5] M. Ciglar. An ultrasound based instrument generating audible and tactile sound. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 19–22, Sydney, Australia, 2010.
- [6] W. contributors. D-beam, 2015. [Online; accessed 27-January-2018].
- [7] W. contributors. Fudi, 2017. [Online; accessed 27-January-2018].
- [8] W. contributors. Laser harp, 2017. [Online; accessed 27-January-2018].
- [9] W. contributors. Theremin, 2018. [Online; accessed 27-January-2018].
- [10] C. Erdem, A. Camci, and A. Forbes. Biostomp: A biocontrol system for embodied performance using mechanomyography. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 65–70, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.
- [11] J. Han and N. Gold. Lessons learned in exploring the leap motion(tm) sensor for gesture-based instrument design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 371–374, London, United Kingdom, 2014. Goldsmiths, University of London.
- [12] A. R. Jensenius, V. G. Sanchez, A. Zelechowska, and K. A. V. Bjerkestrand. Exploring the myo controller for sonic microinteraction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 442–445, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.
- [13] C.-H. Lai and K. Tahiroglu. A design approach to engage with audience with wearable musical instruments: Sound gloves. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.
- [14] A. Mason. Pamela z sculpts musical abstraction from the sounds (and sights) of daily life, 2016. [Online; accessed 27-January-2018].
- [15] K. Nymoen, M. R. Haugen, and A. R. Jensenius. Mumyo - evaluating and exploring the myo armband for musical interaction. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 215–218, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University.
- [16] A. Oulasvirta, T. Roos, A. Modig, and L. Leppänen. Information capacity of full-body movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1289–1298, New York, NY, USA, 2013. ACM.
- [17] J. Paradiso. Electronic music interfaces, 1998. [Online; accessed 27-January-2018].
- [18] S. Şentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg. Crossole: A gestural interface for composition, improvisation and performance using kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.
- [19] S. Serafin, S. Trento, F. Grani, H. Perner-Wilson, S. Madgwick, and T. Mitchell. Controlling physically based virtual musical instruments using the gloves. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 521–524, London, United Kingdom, 2014. Goldsmiths, University of London.
- [20] STMicroelectronics. AN4545: VL6180X basic ranging application note, 6 2014. Rev. 1.
- [21] STMicroelectronics. VL6180X: Proximity and ambient light sensing (ALS) module, 8 2014. Rev. 6.
- [22] STMicroelectronics. STM32F070CB STM32F070RB STM32F070C6 STM32F070F6: ARM-based 32-bit MCU, up to 128 KB Flash, USB FS 2.0, 11 timers, ADC, communication interfaces, 2.4 - 3.6 V, 2 2017. Rev. 3.
- [23] Sweetwater. Roland d-beam sensitivity, 2003. [Online; accessed 27-January-2018].
- [24] D. Tormoen, F. Thalmann, and G. Mazzola. The composing hand: Musical creation with leap motion and the bigbang rubette. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 207–212, London, United Kingdom, 2014. Goldsmiths, University of London.
- [25] C. Traube and J. O. S. III. Estimating the plucking point on a guitar string. In *Proceedings of the COST G-6 Conference on Digital Audio Effects*, Verona, Italy, 2000.
- [26] P. White. Alesis airfx multi-effects processor, 2001. [Online; accessed 27-January-2018].
- [27] M.-J. Yoo, J.-W. Beak, and I.-K. Lee. Creating musical expression using kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 324–325, Oslo, Norway, 2011.