

Haptic Controller Design and Palm-sized Vibrotactile Array

Christoph W. Borst and Charles D. Cavanaugh
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330
{cborst, cdc}@cacs.louisiana.edu

Abstract

We present a haptic controller prototype and a palm-sized vibrotactile array. A primary design goal was to produce a simple device suitable for prototyping ideas or introducing students to haptic controllers. Despite its simplicity, the controller is useful for various applications of low-cost DC tactors. It receives commands from a PC through a standard serial interface and controls up to 32 DC elements such as vibrating pager motors, with intensity of each element controlled by pulse-width modulation. We used the controller to construct a 30-tactor rectangular vibrotactile array as a testbed for feedback to a user's palm using low-cost DC tactors. We describe the use of spatial anti-aliasing for improved sensations. We also introduce a touchpad-driven interpersonal haptic communication system in which the array is driven by touchpad input, thereby allowing one user to haptically "draw" on a remote user's hand.

1. Introduction and Background

Various vibrotactile devices have been developed for indicating contact in virtual environments and for communicating other information haptically. Some examples are vibrotactile gloves and similar devices for virtual environments or telerobotic control [1-4], torso or cockpit displays that provide a natural mapping of orientation cues or warning signals to the body [5-7], chair-mounted back displays for attentional cueing or directional cueing [8], and vibrating controllers that are found on video game systems.

These displays have been built using actuators such as voice coils, miniature loudspeakers, piezo buzzers, or vibrating motors. The use of vibrating DC motors such as those found in cell phones and pagers is becoming increasingly common in tactile displays. They are small, lightweight, inexpensive, consume little power, and various models are now widely available. On the other

hand, they offer limited control over vibration frequency and amplitude since these characteristics depend on the manner in which tactors are mounted or contacted (the behavior of one specific pager motor model is further described in Section 3.2).

Many controllers have been designed for tactile displays. In the case of voice coils or speakers, these typically include function generators and audio amplifiers (e.g., [2, 8]) or may use audio outputs of a soundcard [9]. For the control of DC elements such as pager motors, a general-purpose controller, called the TactaBoard, has been developed by Lindeman et. al. [10, 11]. Each TactaBoard provides control of 16 tactors based on commands received through a standard serial port, and it uses pulse-width modulation (PWM) to implement control of motor intensity.

Some of our basic design goals and high-level design of our controller are similar to those of the TactaBoard. These include developing a simple, compact, low-cost controller for a range of low-power devices. However, we increase emphasis on simplicity and add to this the goal that other researchers should be able to quickly and easily implement and extend our design. Our design uses a Stamp microcontroller that is simple to program and interface with other devices, albeit with a limited execution speed due to its use of an interpreted BASIC-like language. Other differences from the TactaBoard are that our design controls 32 tactors instead of 16, does not require a separate external power supply for tactors, and is an open-hardware design.

Our controller is not the first to use a Stamp. For example, a Stamp was used in the "Hand of Death" [12], which also used the same tactor type found in our tactile array. The device used a Stamp to control a single tactor for bodyworn feedback "pods".

The first version of our controller has been used to implement a 30-tactor vibrotactile array that fits the palm of a large-handed user. An application running on a laptop computer maps touchpad input to array commands so that the array haptically displays the position and intensity of touchpad input.

2. Controller Design

2.1 Hardware Design

Figure 1 provides a high-level overview of our controller design, and Figure 2 shows our initial circuit design for control of 3-Volt motors. Its main components are a Parallax Stamp BS2p40 microcontroller and five ULN2003A darlington transistor arrays. The BS2p40 has 32 I/O pins and two dedicated serial I/O pins, allowing 32 factors to be controlled independently without introducing additional circuitry for output decoding, and allowing the host to connect to the Stamp without tying up I/O pins. The Stamp can source or sink 30mA on each pin with a limit of 60mA as the total for groups of eight pins. Therefore, some additional circuitry is needed to drive factors. The ULN2003A is an inexpensive transistor array that supports current loads of up to 500mA for each output. Each IC has seven logic-level inputs that switch seven outputs, and five such ICs are used in the circuit, giving 35 outputs, of which 32 are used. The ULN2003A has built-in diode clamps that prevent back voltage from the motors from damaging the Stamp.

Figure 3 shows a solderless prototype of the circuit that can be constructed in well under an hour using a project board included with a Stamp kit. A factor battery supply and the factors themselves are the only components not visible in the figure and vary depending on the particular application. A standard serial cable connects to a host computer, and a 2.1 mm power jack connects to a 9-volt power supply. Only three ULN2003A arrays fit on the project board, so only 21 factors are controlled with this prototype.

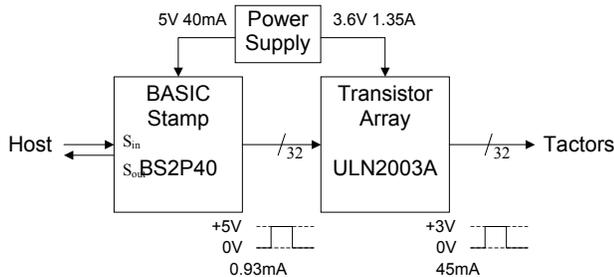


Figure 1: Controller design overview

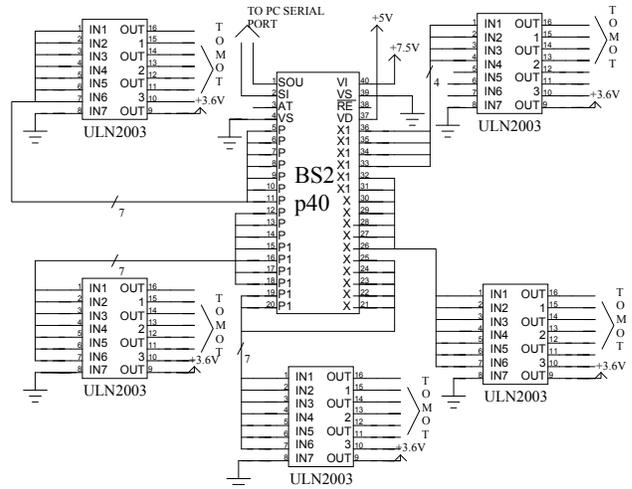


Figure 2: Initial design

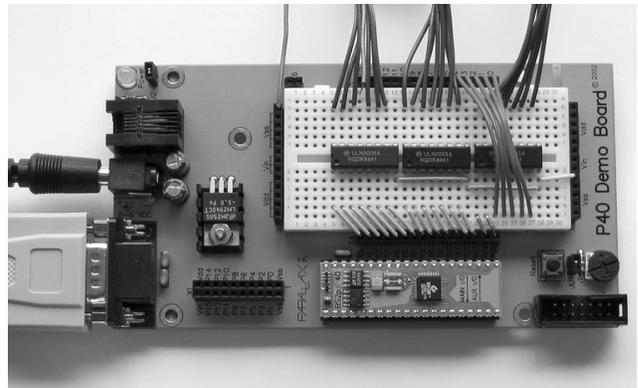


Figure 3: Solderless prototype using Demo Board

Figure 4 shows our controller prototype built using an etched PC board. An external 12VDC 8A switching power supply connects to two voltage regulators (KA7805 and LM350) to provide +5VDC to the Stamp and a transistor array voltage that can be set anywhere in the range +1.25V to +6.75V using an on-board potentiometer. The transistor array has a voltage drop of approximately 0.6VDC, thus array voltage is set to 0.6V greater than the desired factor supply voltage to compensate. Other components consist primarily of filter capacitors for power and serial i/o.

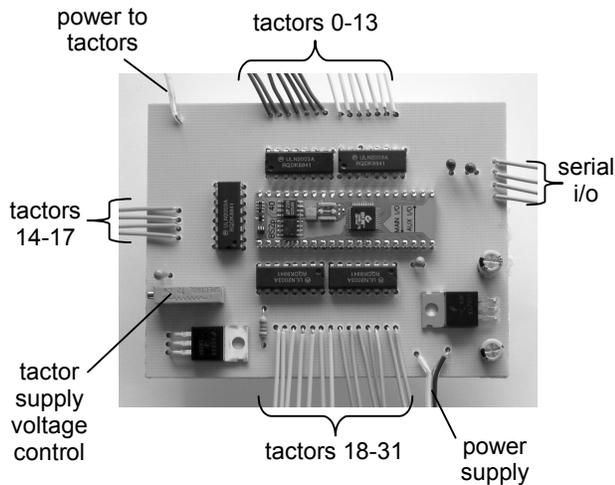


Figure 4: Controller prototype

2.2 Software Design

The controller software consists of a device driver running on a host computer and a tactor controller running on the Stamp. We implemented two software designs. In our first design, the host computer sends one-byte commands to the stamp to indicate any desired changes in state. Each byte consists of a five-bit tactor ID and a three-bit intensity level. The Stamp software then uses pulse-width modulation (PWM) to control the tactors at the desired levels. However, we found that the interpreted BASIC-type language of the Stamp results in slow PWM that produces perceivable pulses when many outputs are controlled with independent levels simultaneously. Therefore, this approach is promising only when the number of tactors activated simultaneously is small.

In the second design, we minimize the size of the Stamp code and move the PWM control to the host. In this case, the host driver controls tactors by continually sending four-byte bit-mapped commands in which each bit corresponds to the current desired state for one of the tactors (on or off). The host driver runs in its own high-priority thread at a rate dependent on the communication rate. We currently use a 56 Kbaud connection to the Stamp, resulting in a stable update rate that is just under 300 Hz (300 four-byte commands per second). The driver spends most of its time waiting for the command bytes to be echoed back by the Stamp. Higher rates may be achieved by ignoring the Stamp's echo, but care must then be taken to control the rate from the driver to not exceed the Stamp's ability to process incoming data.

This allows the state of each tactor to be switched at about 300 Hz, which is fast enough for PWM with a small

number of levels. Before considering this in more detail, we describe the code running on the Stamp.

The simple Stamp code in the second approach is shown in its entirety in Figure 5. Its main loop consists only of statements to read four bytes from the serial port and statements to control the logic level of the corresponding output pins, which are already bit-mapped in OUTH and OUTL registers (high- and low-order bytes).

```
n56000 CON 16408 ` baud rate setting

vall VAR byte

mainio ` set up the pins as outputs
DIRS=%1111111111111111
auxio
DIRS=%1111111111111111

loop:
mainio ` switch output bank
serin 16, n56000, [vall]
OUTL=vall
serin 16, n56000, [vall]
OUTH=vall
auxio ` switch output bank
serin 16, n56000, [vall]
OUTL=vall
serin 16, n56000, [vall]
OUTH=vall
goto loop
```

Figure 5: Stamp code listing

```
In each iteration of the driver loop:

for i = 0 to N-1, where N is the number
of elements controlled

    let pattern be the current control
    pattern for output i, selected
    according to its desired level set by
    an application, with pattern length of
    period control cycles

    set bit i in command to pattern[count
    MOD period], where count counts the
    number of driver iterations passed

end for loop

send the 4-byte command to controller

read 4-byte echo from controller

increment count
```

Figure 6: PWM algorithm

The host driver then uses PWM to control tactor intensity according to desired target levels (tactors such as pager motors do not allow independent control over frequency or amplitude – we use the word *intensity* to describe the controlled effect). PWM techniques control the level of a device by applying a sequence of on/off pulses. Depending on the switching rate and the properties of the device, the result will not be perceived as pulses, but rather as a constant level.

The PWM algorithm is summarized in Figure 6. Each control *period* consists of multiple switching cycles. Each *pattern* describes a level with a one or zero for each switching cycle of a period. Different periods can be used for different levels if desired. For simplicity, we give an example in Table 1 that uses a fixed period of eight switching cycles to achieve nine intensity levels. If a pattern has a one in each position, then the tactor runs at full intensity. If only every other value is a one, then it runs at half intensity, and so on. For our vibrotactile array, we chose 23 levels using variable periods from one to eight switching cycles. Section 3.2 describes the response of a DC tactor to these levels.

An additional feature of the host driver allows an application to set minimum or maximum on times for a tactor. This is useful, for example, if the application wishes to enforce a minimum perceivable pulse duration.

Table 1: Example of eight-cycle patterns

Level	Pattern							
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	0	0	0	1	0	0	0
3	1	0	0	1	0	0	1	0
4	1	0	1	0	1	0	1	0
5	1	1	0	1	1	0	1	0
6	1	1	1	0	1	1	1	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1

3. Vibrotactile Array

3.1 Array Design

We used our controller prototype to construct a vibrotactile array for haptic feedback to the palm of a large-handed user. The array is pictured in Figure 7. The controller is mounted inside a project box and 30 small vibrating motors are arranged in a 5 × 6 grid on top of the box. The motors are mounted on foam pads to help isolate them from the project box and from each other. The foam

also allows the shape of the array to deform slightly according to the shape of the hand.



Figure 7: Vibrotactile array

The motors are Yokyo Parts FM31E (a.k.a. Sanko Electric 1E120) disc-shaped vibrating motors, each 1.4 cm (0.55 in) in diameter and 0.3 cm (0.13 in) thick. These produce a discernable vibration when voltage is applied because of an internal eccentric weight attached to the motor shaft. The tactors are small in size and moving parts are fully enclosed, allowing the user to touch a tactor without concern for interfering with internal components. However, we initially experienced problems with delicate electrical connections and therefore ruggedized the external electrical connections with epoxy drops. The tactors are rated at 3VDC and each draws a current of approximately 45mA. So, with all 30 motors on at the same time, total current draw due to tactors is roughly 1.35A. The power drawn by a motor is $45\text{mA} * 3\text{V} = 135\text{mW}$, creating no noticeable heat.

The array box has a 2.1 mm jack for the external power supply and a female DB-9 serial port for connecting to the host using a standard serial cable. We also included a DB-25 port for driving external tactors. A switch on the side of the box can be used to switch power between array tactors, external tactors, or both in parallel.

3.2 Tactor Response to PWM levels

We measured the response of an array motor to the 23 PWM levels having periods of one to eight switching cycles. In our initial experiment, we placed a telephone pickup coil near the motor and used real-time FFT to observe frequency response to both controller PWM levels and to analog control of levels using a voltage regulator connected directly to the motor. We also varied the manner in which the motor was contacted. This experiment gave us a rough idea of motor behavior. We noted vibration frequencies up to about 100 Hz, large increases in frequency when a motor was contacted firmly rather than allowed to move freely on its foam mount, and an increase in vibration frequency with level that was

roughly similar for PWM control and direct control of voltage.

Our second experiment measured motor behavior more precisely. We sandwiched the motor between two foam pads to simulate hand contact, verifying that the vibration frequency at the maximum level was identical to that typical during real hand contact. We placed a RODE NT5 condenser microphone above this arrangement (the manufacturer reports a flat low-end frequency response for this microphone) and recorded frequency and magnitude of the fundamental peak from a real-time FFT plot of the microphone signal. The response to our 23 PWM levels is shown in Figure 8 and Figure 9. Average voltage is the product of constant supply voltage and PWM duty cycle.

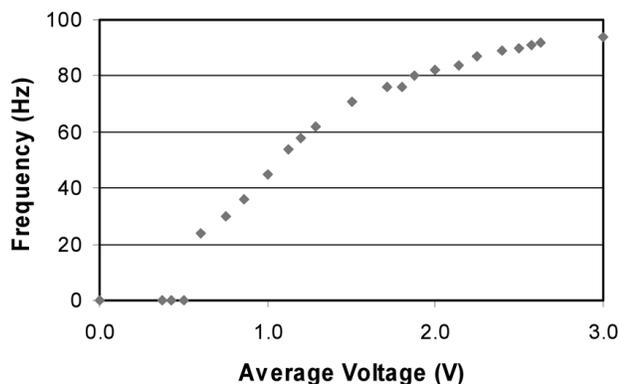


Figure 8: Response to PWM levels (frequency)

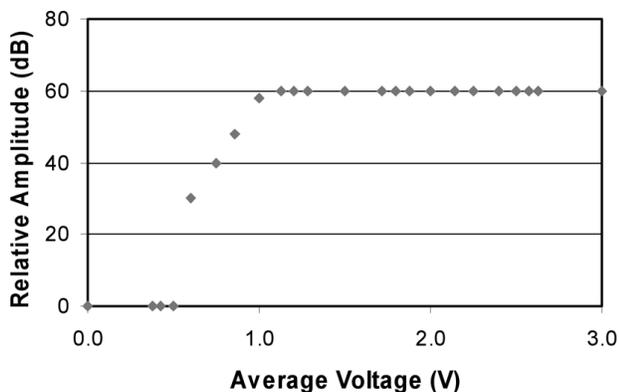


Figure 9: Response to PWM levels (amplitude)

The plots suggest a logarithmic increase in vibration frequency as duty cycle increases. Vibration frequency range is (roughly) 20 Hz to 100 Hz but depends on the manner in which a motor is contacted (contact was fixed

for this experiment). During the experiment, we also noted frequency variation around $\pm 5\%$ even with constant control conditions. The amplitude plot suggests there was no increase in amplitude for average voltage beyond about 1.0V. Further investigation is needed to explain this result. We verified that the measurement apparatus was able to pick up higher-amplitude vibrations without visible signs of clipping in the signal waveform.

The use of pager motors is attractive due to low cost and simplicity of implementation, but vibration frequency and amplitude cannot be controlled independently and neither parameter can be controlled precisely. However, a growing body of research demonstrates the potential value of such factors.

3.3 Anti-Aliasing

The limited array resolution results in perceivable spatial aliasing artifacts. For this reason, we implemented spatial anti-aliasing using the multiple motor levels, analogous to spatial anti-aliasing in graphical displays using multiple pixel intensities. Temporal anti-aliasing may also be useful, and would result in the haptic counterpart of visual motion blur.

Consider haptic representation of a single point coordinate on the hand coordinate system. Without anti-aliasing, such a point can be represented by activating the nearest motor at fixed intensity, and then the resolution for display of coordinates is limited to tactor resolution (here, 5×6). In case a moving point is represented with this approach, discrete jumps occur as a motor is turned off and its neighbor is turned on. We implemented a simple spatial anti-aliasing function that distributes the motor intensity among up to four nearby motors, with individual motor intensity proportional to distance between motor center and the represented coordinate. This reduces perceivable discrete changes, potentially producing smoother sensations of motion and allowing sub-tactor changes to be felt. Formal evaluation is needed to investigate the perceptual effects.

We also implemented two anti-aliasing techniques for lines. We display a line as a moving point, so a straightforward anti-aliasing technique simply applies the single-point technique to the moving point. A second technique again uses a moving point, but only distributes intensity among the two closest motors that would be activated if the array were treated as a raster by a classic line-drawing algorithm such as Bresenham's [13].

It is not yet clear how effective anti-aliasing can be when the intensity levels correspond to largely different frequencies and not just amplitudes. This also raises concerns about beats (low-frequency pulses that can occur when multiple motors vibrate simultaneously at different frequencies). Anecdotally, the problem of beats has been minor with our array and may be outweighed by the

benefit of simultaneously activating multiple motors for anti-aliasing, at least for certain applications. Again, formal evaluation is needed.

3.4 Array Applications

The vibrotactile array is potentially useful for a variety of applications, but further work is first needed to evaluate basic capabilities. The device may be used to haptically display images, directional cues, or other information to a user. For example, a user interacting with an environment using a one-handed input device may place the other hand on the tactile array to receive haptic information about the environment in addition to any visual feedback already present. The array can also be integrated with an input device to allow a hand to simultaneously provide input and receive haptic feedback. Our haptic controller is also useful for other configurations of low-cost DC tactors, such as haptic gloves or vests.

One application we consider is the use of the array for interpersonal haptic communication. Previous haptic devices for interpersonal communication include the ComTouch vibrotactile device for augmenting voice communication [14], the inTouch mechanical rollers that allow manipulation of a seemingly shared device [15], and the HandJive hand-held entertainment device [16], among others. Such devices can establish a sense of co-presence and communicate emotion or other information. We use the array to approximate the sensation of a user's fingertip on another user's palm, thereby allowing one user to haptically "draw" on a remote user's hand. Figure 10 illustrates the touchpad-driven interpersonal haptic communication concept. A user touches a sensing surface, in this case a laptop touchpad, that measures location and intensity of contact. The contact coordinate is mapped to the vibrotactile array's coordinate system, and the spatial anti-aliasing function distributes intensity among up to four nearby motors.

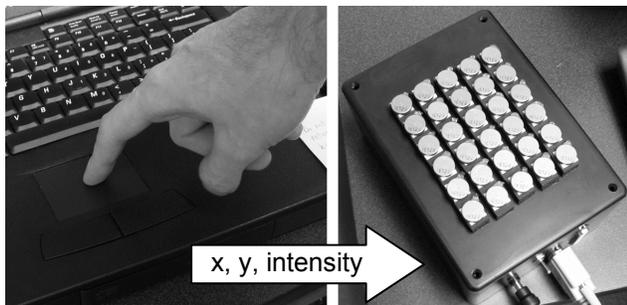


Figure 10: Touchpad-driven interpersonal haptic communication

In addition to touchpad-driven communication, we have used the array to display direction vectors, shapes, and other information. Our initial reaction is that static "images" are difficult to interpret but that motions are understandable.

4. Conclusion and Future Work

We designed a controller for controlling 32 devices such as the vibrating motors that are increasingly common in vibrotactile displays. The controller design is simple and can easily be duplicated or extended by other researchers. Its main limitation is that the relatively slow execution of instructions on the Stamp limits the ability to control levels independently for all 32 tactors from the microcontroller. This may be remedied with faster versions of the Stamp in the future or with a more sophisticated microcontroller and a separate peripheral I/O controller. However, use of a Stamp resulted in a simple controller design, and levels can be implemented for our tactile display by moving PWM control to the host driver.

We used the controller to construct a 30-element vibrotactile array for feedback to a user's palm. We described the use of spatial anti-aliasing to produce improved sensations and presented the concept of touchpad-driven interpersonal haptic communication.

Although formal evaluation of the system has not yet been conducted, we received useful feedback from several users during a recent demonstration at a major haptics symposium. Most users found the mapping from touchpad to palm to be natural and commented positively on the effect. Some users commented that the flat arrangement of motors was not ideal for the contour of a hand. Although the foam mounting allowed the array to deform slightly to users' hands, a dome-like arrangement would have improved consistency of contact between motors and the palm surface. Some small-handed users could not contact all 30 motors at once, and some users preferred to contact the array with their fingers or a forearm. When presented with the ability to toggle options such as anti-aliasing and sensing of contact intensity by the touchpad, preferences varied. Comments about the anti-aliasing option suggested that the resulting sensations were smoother and that motor level appeared increased.

We are currently working to improve the system and conduct formal evaluations. We plan to improve feedback by calibrating motor levels to a perceptual scale and by optimizing the PWM patterns to provide the greatest number of useful levels without introducing perceivable pulses. Further development of anti-aliasing techniques for tactile arrays and evaluation of their perceptual effects are other interesting areas for future work.

Information about the array is currently maintained at <http://www.cacs.louisiana.edu/~cborst/tactilearray/>.

Acknowledgements

The authors thank David Pope and Arun P. Indugula for their work on device drivers and demo programs during project courses.

References

- [1] Virtual Technologies, "VTi Product Brochure." Palo Alto, CA: Virtual Technologies, Inc., 1999.
- [2] N. Patrick, "Design, Construction, and Testing of a Fingertip Tactile Display for Interactions with Virtual and Remote Environments," Master's Thesis, Department of Mechanical Engineering. Cambridge, MA: MIT, 1990.
- [3] D. Kontarinis and R. Howe, "Tactile Display of Vibratory Information in Teleoperation and Virtual Environments," *Presence: Teleoperators and Virtual Environments*, vol. 4, pp. 387-402, 1995.
- [4] A. Kron and G. Schmidt, "Multi-fingered Feedback from Virtual and Remote Environments," presented at 11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Los Angeles, 2003.
- [5] J. L. Rochlis and D. J. Newman, "A Tactile Display for International Space Station (ISS) Extravehicular Activity (EVA)," *Aviation, Space, and Environmental Medicine*, vol. 71 (6), pp. 571-578, 2000.
- [6] H. van Veen and J. van Erp, "Tactile Information Presentation in the Cockpit," presented at Haptic Human-Computer Interaction, Glasgow, UK, 2000.
- [7] J. van Erp and H. van Veen, "A Multi-purpose Tactile Vest for Astronauts in the International Space Station," presented at Eurohaptics 2003, Dublin, Ireland, 2003.
- [8] H. Z. Tan, R. Gray, J. J. Young, and R. Traylor, "A Haptic Back Display for Attentional and Directional Cueing," *Haptics-e* <http://www.haptics-e.org>, vol. 3, 2003.
- [9] P. Wellman and R. Howe, "Towards Realistic Vibrotactile Display in Virtual Environments," presented at Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 1995.
- [10] R. W. Lindeman, J. N. Templeman, J. L. Sibert, and J. R. Cutler, "Handling of Virtual Contact in Immersive Environments: Beyond Visuals," *Virtual Reality*, vol. 6, pp. 130-139, 2002.
- [11] R. W. Lindeman and J. R. Cutler, "Controller Design for a Wearable, Near-Field Haptic Display," presented at 11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Los Angeles, California, 2003.
- [12] T. E. von Wiegand and J. D. Pfautz, "The 'Hand of Death,' a distributed haptic interface for signaling exposure to virtual enemy fire," *Presence-Connect* <http://www.presence-connect.com>, vol. 24/09/2001, 2001.
- [13] J. E. Bresenham, "Algorithm for Computer Control of a Digital Plotter," *IBM Systems Journal*, vol. 4, pp. 25-30, 1965.
- [14] A. Chang, S. O'Modhrain, R. Jacob, E. Gunther, and H. Ishii, "ComTouch: Design of a Vibrotactile Communication Device," presented at Designing Interactive Systems, London, England, 2002.
- [15] S. Brave and A. Dahley, "inTouch: A Medium for Haptic Interpersonal Communication," presented at Conference on Human Factors in Computing Systems, Atlanta, Georgia, 1997.
- [16] B. Fogg, L. D. Cutler, P. Arnold, and C. Eisbach, "HandJive: A Device for Interpersonal Haptic Entertainment," presented at Conference on Human Factors in Computing Systems, Los Angeles, California, 1998.