Enhancing VR-based Visualization with a 2D Vibrotactile Array

Christoph W. Borst*
Center For Advanced Computer Studies
University of Louisiana, Lafayette

Vijay B. Baiyya†
Center For Advanced Computer Studies
University of Louisiana, Lafayette

Abstract

We discuss methods to enable haptic visualization on vibrotactile arrays. Our work is motivated by the potential for a tactile array to provide an additional useful channel for information such as location cues related to dataset features or remote user behaviors. We present a framework for array rendering and several specific techniques. Novel aspects of our work include the example application of a palm-sized tactile array to visualize dataset features or remote user state in a VR system, a generalized haptic glyph mechanism for 2D tactile arrays, and the extension of graphical visualization techniques to haptics (glyphs, fisheye distortion, spatial anti-aliasing, gamma correction).

CR Categories: H.5.2 [Information Interfaces And Presentation (e.g., HCI)]: User Interfaces—Haptic I/O; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality;

Keywords: Vibrotactile Array, Haptics, Tactile Map, Haptic Exploration, Haptization, Haptic Glyphs

1 Introduction

In this paper, we describe methods for vibrotactile array rendering to enhance data exploration. Our work is applicable to various 2D array configurations, but we illustrate with the example of a low-cost palm-sized vibrotactile array to provide information about location and behavior of remote users in a collaborative VR system, i.e., a tactile version of a heads-up map. This is realized in part by the development of a "haptic glyph" mechanism with several parameters that can be varied to communicate desired information. This paper builds on our earlier work to develop and evaluate hardware and low-level rendering methods for 2D vibrotactile arrays [Borst and Asutay 2005] and integrates ideas such as haptic glyphs [Roberts and Franklin 2005] and distorted views [Leung and Apperley 1994] applied in a new way to enhance haptic display.

2 Motivation and Example Application

The use of low-cost vibrotactile elements (tactors) is increasingly common in haptics research and applications. Small vibrating DC motors and Piezo speakers have been used to construct haptic displays with low power requirements, low cost, low weight, portability, and simple implementation. Commonly-explored configurations include elements placed in clothing or used to build arrays on the backs of seats to communicate shape, orientation, direction, or attention cues for both real and virtual environments, e.g., [Yanagida et al. 2004; Tan et al. 2003; van Veen and van Erp 2001; Rochlis and Newman 2000; Toney et al. 2003; Lindeman et al. 2005; Bloomfield and Badler 2007]. Experimental studies have verified that such cues can be communicated.

*e-mail: cborst@cacs.louisiana.edu
†e-mail:vbb4359@cacs.louisiana.edu

VRST 2007, Newport Beach, California, November 5–7, 2007.
© 2007 ACM 978-1-59593-863-3/07/0011 $5.00

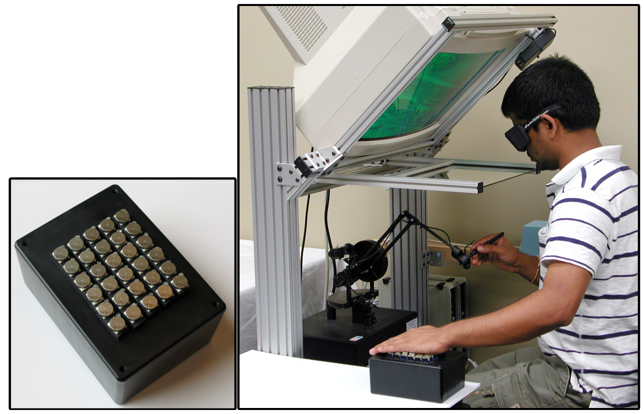

Figure 1: *Left: Example of a vibrotactile array. Right: Vibrotactile array in fishtank-type VR system.*

Figure 1 shows a 2D vibrotactile array prototype being used in a geosciences application that we have presented elsewhere [Borst et al. 2006]. The relevant aspects of the application are that the user studies multiple datasets from the same geographic region for interpretation, and that the application is networked with a remote system for collaborative exploration. Besides the vibrotactile array, the hardware consists of a mirror-based fishtank-style VR display with a Phantom force-feedback stylus as the interaction tool.

We believe that the introduction of a palm-sized tactile array into this type of VR display provides an additional useful channel for information, especially location cues, with minimal added cost or cumber. While the Phantom device can provide contact cues for stylus interactions and other force cues, earlier-cited research suggests a vibrotactile array is suited to location cues, which do not map so directly to the Phantom. Besides our VR setting, tactile arrays can be used with conventional 2D desktops, can be built in various form factors, and might one day be used in other applications such as cues for vehicle operators. Smaller tactors that are already available would allow our array to be made smaller or denser and possibly adapted to mount on a mouse or hand-held controller.

Using the array as a tactile map of remote users' positions and actions addresses a problem that is particularly notable in fishtank VR displays due to their small field of regard: avatars or view proxies conveying similar information can be difficult to find or keep track of when not immediately in front of the user. Even if visual indicators (such as a graphical heads-up map) are kept in view, they reduce space available for dataset views and interactive objects, and may distract visual attention from the task being performed.

3 Array Prototype and Capabilities

Our vibrotactile array prototype consists of 30 vibrating motors on a controller box with a serial connection to a PC. The tactors (Sanko Electric IE120) form a 5×6 grid with grid spacing of about 18mm and are mounted on foam pads to help isolate them from each other and to help the array conform to hand shape. We measured the fundamental vibration frequency of tactors to range from 27 Hz to 100 Hz, depending on contact pressure and input. For further details, see [Borst and Cavanaugh 2004].

At the lowest level, the controller treats the array as a monochrome raster (30 bits). However, we use pulse-width modulation (PWM) with a switching rate of roughly 300 Hz to provide 23 tactor intensity levels, allowing applications to treat the array as a grayscale raster. Furthermore, driver software provides a function for rendering point primitives expressed with respect to an array coordinate system. Line segments or curves are rendered as moving point traces, because static raster images spanning several motors are more difficult to perceive meaningfully (e.g., see discussion in [Yanagida et al. 2004]). A trace is generated by evaluating a parametric equation describing a point on a curve as a function of time. A conversion of the point to actual tactor intensities, described next, allows an application designer to focus on generating point or curve primitives rather than specific tactor activation sequences.

A distinguishing feature of our array rendering approach is our use of spatial anti-aliasing and gamma correction, as detailed in [Borst and Asutay 2005]. We use unweighted area sampling to compute tactor intensities for a point primitive, i.e., the intensity for each tactor is the area of overlap between its unit-area grid cell and a unit square centered on the rendered point coordinate (we also developed an alternative rendering method, "interpolated midpoint", for tracing shapes smoothly; see [Borst and Asutay 2005]). Any nonzero intensity I is furthermore adjusted by the function:

$$I_{adjusted} = (1-\mu)(I)^{\frac{1}{\gamma}} + \mu, \tag{1}$$

where μ is a threshold parameter that sets minimum nonzero tactor intensity (e.g., minimum level resulting in vibration) and γ is a gamma parameter that can adjust for nonlinearities in tactor or perceptual responses (or can be tuned to preference). Our driver selects the PWM pattern having duty cycle closest to $I_{adjusted}$.

We previously investigated these techniques using human factors and psychophysics methods. We showed that anti-aliasing, by either area sampling or an interpolated midpoint technique, improves perceived quality. And, we showed that the combination of area sampling and proper selection of μ and γ allows users to detect the direction of short line segments rendered on the array, even for line segments with length below one grid spacing. This result and results from earlier-cited research suggest that low-cost tactor arrays are usable for location or direction cues rendered as points or line traces.

On the other hand, some limitations of this array type should be noted. Resolution of our prototype is low and the low-cost DC tactors do not allow precise or independent control of vibration frequency and amplitude. Shape discrimination is difficult when it hinges on detection of corners in traced shapes (e.g., square vs. circle [Borst and Asutay 2005]). Beats (low-frequency pulses) can occur when multiple tactors vibrate simultaneously at different frequencies, but our previous experiments nonetheless showed quality improvements for anti-aliased approaches that use multiple tactors. Perceivable beats are minimized by avoiding a static or slow-moving stimulus, so any remaining beats are transient. Finally, motors can have a low-pass filtering effect on intensity profiles of our glyph mechanism (presented in Section 4.1.2), limiting the range of possible effects. Note that the severity of these limitations depends on the particular tactor and control technology used, but the focus of this paper is on concepts that can generalize to other arrays rather than on characteristics of a specific array design.

4 Supporting Haptic Visualization

We built on our earlier work by developing higher-level mechanisms to support haptic visualization (sometimes called haptization or haptification) of information for applications such as our geosciences application. The main extension to our framework is the development of a haptic glyph mechanism that matches well to the array's capabilities.

4.1 Haptic Glyph Method

4.1.1 Glyph Concept and Related Work

A glyph is an object that is modified by input data to communicate information. In graphical visualization, glyphs have been described as types of icons (e.g., [Delmarcelle and Hesselink 1994]), but Roberts and Franklin [2005], who presented haptic glyphs, distinguished them from icons by explaining that glyphs actively encode information mapped to multiple parameters, while icons have constant form and unique association. Based on this, work on haptic glyphs is minimal but is related to techniques that have been called haptic icons and tactons. These techniques have been applied primarily to single-transducer systems. Roberts and Franklin briefly proposed force glyphs for the Phantom using grooves and caverns [2005], while Osawa investigated sequence and strength patterns for Cybertouch glove-mounted tactors to represent abstract information (calling them tactile glyphs) [2006]. Enriquez and MacLean introduced editable force profiles called haptic icons for a force-feedback knob [2003], which led to later investigations of haptic icons for other transducers (e.g., [Chan et al. 2005]). Brewster and Brown describe tactons and potentially useful parameters such as frequency, amplitude, and duration, which could represent different properties [2004]. Their later work used up to three transducers, spaced apart on an arm. None of these works provided a general glyph mechanism or rendering framework for arrays such as ours.

4.1.2 General Haptic Glyph for 2D Vibrotactile Arrays

Our glyph mechanism allows an application to generate a glyph by specifying any subset of the following parameters:

Shape: The shape to render, specified by a list of curve segments. For example, a linear segment is specified by two 2D endpoints P_0 and P_1, and the system traces this segment by varying t from 0 to 1 in $(P_0 + (P_1 - P_0)t)$ according to a timer. This approach extends readily to standard curve types, such as Bézier segments or 2D NURBS. If no shape is specified, the system defaults to using a point (the origin).

Position: Specifies a 2D translation applied to the glyph, or zero by default.

Orientation: Specifies a rotation (one angle) applied to the glyph, or zero by default. The rotation is applied in the glyph's local coordinate frame, which may be translated relative to the array frame.

Scale (2D): Specifies a 2D scale factor, or no scaling by default. This transformation acts in the glyph's local frame, which may be translated and rotated relative to the array frame.

Count: Specifies how many more times the shape should be traced, with a default of the maximum representable value.

Durations: Specifies timing for shape tracing and is used to compute t-values needed to trace segments. Two different forms are supported for this specification: A pair of values can be given to specify total shape trace time and a delay following the trace, or a list of pairs can be given to specify this on a per-segment basis. Default value is one second per-shape trace time and no delay.

Intensity Profile: When specified, this is a profile used to modulate tactor intensity. It can be specified on a per-shape, per-segment, or real-time basis. An intensity profile consists of a list of time-intensity pairs (linear interpolation generates intermediate values). For per-shape or per-segment specification, time values are multiplied by duration so they are specified in normalized form. For real-time specification, values are in seconds with no normalization, and additional offset and repeat values are available to support repeating patterns or haptic icons arriving at arbitrary times for immediate display. Finally, multiple profiles may be specified in a priority order to support composite haptic icons or a return to previous behavior after icon rendering.

There are two different glyph update methods. First, the application programmer can specify a callback function that will be called at the end of a glyph's tracing cycle, allowing the application to update parameter values before the next trace. Alternatively, provided functions allow an application to update parameters at any time, but care must be taken to avoid undesirable side effects such as glyphs feeling discontinuous when position is changed.

Multiple glyphs can be displayed, but this is only done serially, since simultaneous display is likely to make them uninterpretable.

These parameters support a wide range of effects and, together with the methods summarized in Section 3, provide a unified framework for rendering cues on 2D tactor arrays.

4.2 Dataset Feature Extraction

Methods are needed for mapping information about VR environments to glyph parameters. In some cases, such as the collaborative map discussed in Section 5, information is already in a form that readily maps to glyph parameters. In other cases, such as haptic representation of terrain data features, information must be converted to a suitable form. For example, in our geosciences application, extrema and crevices in topographic data (e.g., SRTM and LIDAR meshes) are studied and compared to features in associated geophysical datasets (e.g., gravity magnitude). A 2D haptic display may be useful for conveying properties such as position, shape, magnitude, or direction for such features when not directly visible. For example, a feature of interest may be associated with a secondary dataset, may not be illustrated by current visual parameters, or may be out of view due to scaling or viewpoint.

Currently, an extrema detection mechanism considers a dataset subregion inside a region selector that follows the user's interaction tool during exploration, i.e., a region near the Phantom tool tip. The data may be from a secondary dataset that differs from visually-displayed data, and the size of the region selector can be adjusted and may correspond to a much different scale than that seen in the visual display. The mechanism finds minima or maxima in the selector range and extracts coordinates and values for mapping to a glyph. Several mappings are possible. Currently, we map coordinates to a point glyph's position. This is a simple example; this work can be extended with more advanced feature extractions and mappings to glyph parameters.

4.3 Distorted Haptic View

We apply a distortion technique from graphical visualization to haptic visualization to address the following problem: If a tactor array represents a large region of an environment or dataset, limited detail is sensed due to limited resolution of the array and of corpuscles that sense vibration. And, if an array represents a small region, important information outside that region may be missed.

We address this by distorting glyph positions according to a fisheye view technique. This allows glyph positions near the center of the array to convey position of mapped features near the user or a region of interest, while reduced detail near the array edges only allows glyph position to communicate direction. Of course, the fisheye distortion is optional and glyph parameters other than position could be used to further communicate or reinforce position information. The specific form of distortion we use is:

$$T(x) = \frac{(d+1)x}{(dx+1)}, \tag{2}$$

where d is a distortion factor and x is a 1D coordinate expressed in a canonical array-centered coordinate system (the equation is applied twice: once for each dimension of the 2D array coordinate). This distortion form, along with several other potentially useful view distortions from graphical visualization, is surveyed by Leung and Apperley [1994].

5 Example: Tactile Map for Collaboration

As motivated in Section 2, we believe our techniques are suited for haptically conveying information about remote users during collaboration. Figure 2 summarizes just one of several ways this can be accomplished with our haptic glyph mechanism. The main approach is to map position of a remote user to haptic glyph position as a location cue. Here, position may refer to head or body position, interaction tool position, or a view target, depending on intent. To create the haptic version of a heads-up map (forward-up map, as opposed to fixed-orientation map), the local user's pose (position and orientation) is also used, to convert remote user pose to relative form. In addition to the location cue, this relative pose is used to provide orientation of the remote user by controlling glyph orientation, with glyph shape being a short line segment as an orientation vector. The fisheye transform is applied to glyph position to provide reasonable position resolution for a virtually-nearby collaborator while providing only a sense of direction for a virtually-distant collaborator. Distance could additionally be mapped to other glyph parameters, such as scale or duration, to help address the limitation for large distances.

Discrete actions of the remote user are mapped to intensity profiles by using event IDs to index profiles stored in a lookup table (LUT). Examples of events include a user joining or leaving the session, placing an interpretive mark on a dataset, or switching views between multiple co-located datasets. Repeating pulses or patterns could be used to continually remind a user which of the co-located datasets the collaborator is exploring, potentially reducing confusion about differences in view. As suggested by existing work on haptic icons, intensity profiles may also be effective for mediating turn-taking for collaboration without intrusive visual indicators [Chan et al. 2005]. For example, one-shot profiles could indicate gain or loss of control, while periodically-displayed profiles could remind a user of pending requests for control. To contrast these examples, a much simpler use of intensity profiles is to simply map information to an intensity level that is constant per glyph trace,

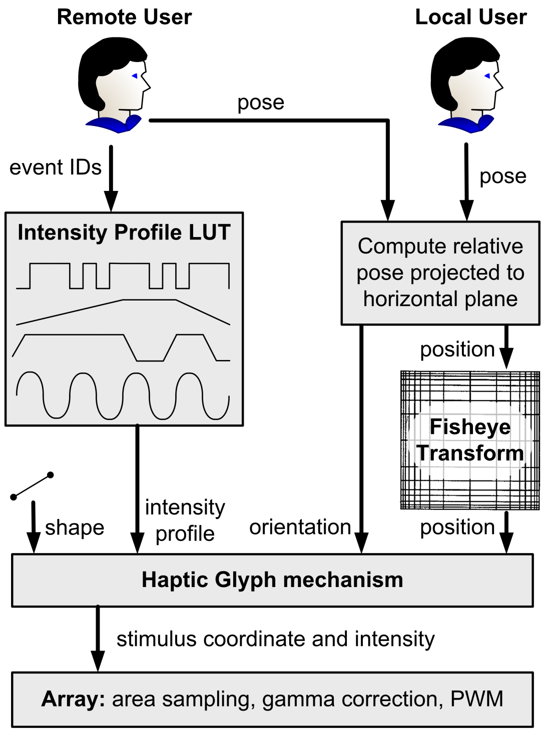

Figure 2: *Summary of one possible approach to a 2D haptic map for awareness of a collaborator's pose and actions.*

e.g., to represent virtual distance between users. The detailed design of intensity profiles is not within the scope of this paper, but has begun to be addressed in the cited works. The focus of our presented work has been to develop the basic methods to provide general support for haptic visualization on 2D tactor arrays.

6 Conclusion and Future Work

We developed a framework for haptic visualization on vibrotactile arrays, along with several specific methods, including a haptic glyph mechanism. We illustrated our techniques with a discussion of a haptic collaborative map for providing awareness of a remote user's pose and actions in a collaborative system, and also briefly discussed extraction and mapping of dataset features to an array.

Future work will focus on user-based evaluation of specific techniques and on development of a second-generation array. Evaluation is important for understanding what specific types of information can be communicated effectively and how to best map information to haptic glyphs. Smaller tactors that are already available will allow us to increase array density or reduce size of the array, and possibly to mount it on a mouse or in a worn sleeve.

References

BLOOMFIELD, A., AND BADLER, N. I. 2007. Collision Awareness Using Vibrotactile Arrays. In *Proceedings of IEEE VR*, IEEE Computer Society Press, 163–170.

BORST, C. W., AND ASUTAY, A. V. 2005. Bi-Level and Anti-Aliased Rendering Methods for a Low-Resolution 2D Vibrotactile Array. In *Proceedings of WorldHaptics*, IEEE Computer Society Press, 329–335.

BORST, C. W., AND CAVANAUGH, C. D. 2004. Haptic Controller Design and Palm-sized Vibrotactile Array. Tech. rep., Center for Advanced Computer Studies, University of Louisiana, Lafayette.

BORST, C. W., KINSLAND, G. L., BAIYYA, V. B., GUICHARD, A. M., INDUGULA, A. P., ASUTAY, A. V., AND BEST, C. M. 2006. System for Interpretation of 3D Data in Virtual Reality Displays and Refined Interpretations of Geophysical and Topographic Data from the Chicxulub Impact Crater. In *Gulf Coast Association of Geological Societies*, 87–100.

BREWSTER, S., AND BROWN, L. M. 2004. Tactons: Structured Tactile Messages for Non-Visual Information Display. In *AUIC '04: Proceedings of the Fifth Conference on Australasian User Interface*, Australian Computer Society, Inc., 15–23.

CHAN, A., MACLEAN, K., AND MCGRENERE, J. 2005. Learning and Identifying Haptic Icons under Workload. In *Proceedings of WorldHaptics*, IEEE Computer Society Press, 432–439.

DELMARCELLE, T., AND HESSELINK, L. 1994. *A Unified Framework For Flow Visualization*. CRC Press, ch. 5, Computer Visualization Graphics Techniques for Scientific and Engineering Analysis, R. S. Gallagher (Ed.), 129–170.

ENRIQUEZ, M. J., AND MACLEAN, K. E. 2003. The Hapticon Editor: A Tool in Support of Haptic Communication Research. In *Proceedings of HAPTICS*, IEEE Computer Society Press, 356–362.

LEUNG, Y. K., AND APPERLEY, M. D. 1994. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Trans. Comput.-Hum. Interact. 1*, 2, 126–160.

LINDEMAN, R. W., SIBERT, J. L., MENDEZ-MENDEZ, E., PATIL, S., AND PHIFER, D. 2005. Effectiveness of Directional Vibrotactile Cuing on a Building-Clearing Task. In *Proceedings of CHI '05*, ACM Press, 271–280.

OSAWA, N. 2006. Tactile Glyphs for Palpation of Relationships. In *Proceedings of the Conference on Information Visualization*, IEEE Computer Society Press, 575–584.

ROBERTS, J. C., AND FRANKLIN, K. 2005. Haptic Glyphs (Hlyphs) - Structured Haptic Objects for Haptic Visualization. In *Proceedings of WorldHaptics*, IEEE Computer Society Press, 369–374.

ROCHLIS, J., AND NEWMAN, D. 2000. A Tactile Display For International Space Station (ISS) Extravehicular Activity (EVA). *Aviat Space Environ Med 71*, 6, 571–578.

TAN, H. Z., GRAY, R., YOUNG, J. J., AND TRAYLOR, R. 2003. A Haptic Back Display for Attentional and Directional Cueing. *Haptics-e: The Electronic Journal of Haptics Research 3*, 1 (June).

TONEY, A., DUNNE, L., THOMAS, B. H., AND ASHDOWN, S. P. 2003. A Shoulder Pad Insert Vibrotactile Display. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, IEEE Computer Society Press, 35–44.

VAN VEEN, H. A. H. C., AND VAN ERP, J. B. F. 2001. Tactile Information Presentation in the Cockpit. In *Proceedings of the First International Workshop on Haptic Human-Computer Interaction*, Springer-Verlag, 174–181.

YANAGIDA, Y., KAKITA, M., LINDEMAN, R. W., KUME, Y., AND TETSUTANI, N. 2004. Vibrotactile Letter Reading Using a Low-Resolution Tactor Array. In *Proceedings of HAPTICS*, IEEE Computer Society Press, 400–406.