

---

# Artex: Artificial Textures from Everyday Surfaces for Touchscreens

**Andrew Crossan, John Williamson and Stephen Brewster**

Glasgow Interactive Systems  
Group  
Department of Computing Science  
University of Glasgow  
Glasgow, UK, G12 8QQ, UK  
{ac, jhw, stephen}@dcs.gla.ac.uk

## Abstract

The lack of tactile feedback available on touchscreen devices adversely affects their usability and forces the user to rely heavily on visual feedback. Here we propose texturing a touchscreen with virtual vibrotactile textures to support the user when browsing an interface non-visually. We demonstrate how convincing pre-recorded textures can be delivered using processed audio files generated through recorded audio from a contact microphone being dragged over everyday surfaces. These textures are displayed through a vibrotactile device attached to the back of an HTC Hero phone varying the rate and amplitude of the texture with the user's finger speed on the screen. We then discuss our future work exploring the potential of this idea to allow browsing of information and widgets non-visually.

## Keywords

Touchscreen, vibrotactile feedback, mobile interaction.

## ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]:  
User Interfaces: *Haptic I/O*.

## Introduction

Touchscreen interactions are fast becoming the norm for many types of user interfaces. The high end mobile

---

Copyright is held by the author/owner(s).

*CHI 2010*, April 10–15, 2010, Atlanta, Georgia, USA.

ACM 978-1-60558-930-5/10/04.

phone market has seen a large scale shift towards capacitive or resistive touchscreens. Further to this, touchscreen devices, particularly since the release of the Apple iPhone, have seen a shift away from stylus-based to finger-based interaction. This shift towards touch interactions is also now happening on the desktop with multitouch tabletop computers becoming more widely available (e.g. Microsoft's Surface device). Touchscreens provide many benefits over more traditional physical button based interactions. They provide the developer with the means to integrate direct manipulation and simple gestures into the interface to allow the user to scroll with the flick of a finger, change scale or rotate easily. This makes them ideal for tasks such as Web or image browsing where the user may wish to take a non-linear path through the data or navigate through a large data space on a small screen.

However, touchscreens have some disadvantages compared to physical button-based interactions. They are necessarily flat, removing all tactile feedback from the interactions that are inherently present when interacting with physical buttons. This makes them virtually inaccessible without the use of the visual channel. This is an issue both for visually impaired users as well as users on the move whose visual attention is focused on safe navigation. This can also lead to inconvenient interactions as users typically must remove the device from their pocket before interacting, whereas physical buttons allow tactile exploration and operation of a device without any visual attention. Physical buttons support a 'scan mode' that allows the user to feel the different physical button shapes, textures and locations without activating any functionality. An example where this proves useful is for keyboard based interactions where, on English keyboards, the 'f' and 'j' keys have

bumps that allow users to identify them and orient their fingers on the keys without looking at the keyboard, aiding touch typists.

For most touchscreen interactions, a touch on the screen will activate functionality. The user cannot therefore explore the controls without interacting with an application. There is a body of work that shows that the lack of tactile feedback can also lead to slower and more error prone interactions. An experienced typist on a standard QWERTY keyboard can reach well over 60 words per minute, which compares with a predicted rate of around 43 words per minute for an experienced touchscreen typist [6]. This is partly due to the 'fat finger problem' where the user's finger obscures the target as he or she attempts to press on it. However, there is also evidence to suggest that the lack of tactile or auditory feedback from pressing the onscreen targets degrades performance [3]. The users do not get the physical cues from the buttons (the movement of the button or the click for a successful selection) which indicates whether a successful selection has been made as the flat nature of the screen means that the user gets no feedback on the edges of buttons.

Both tactile and auditory feedback have previously been used as mechanisms to alleviate some of the usability problems experienced with touchscreen devices (eg. [4, 7, 9, 11]). This is particularly the case for interaction with components such as buttons or sliders where the tactile feedback is used to indicate an interaction event. Nashel and Razaque use tactile feedback to tell users when they are over touchscreen buttons alerting them to button entry and when hovering over the button [7]. Researchers have previously looked at how to encode information in this tactile sig-

nal to provide more information than a simple buzz. For example, Hoggan *et al.* [3] examine interaction with a touchscreen keyboard. They use different tactile feedback for the 'f' and 'j' than for the other keys as it allows users to locate and orient themselves as with a physical keyboard. Different tactile feedback is used for button click and slip off events to alert the user when errors have been made. Hoggan *et al.* were able to demonstrate significant improvements in typing performance on a flat touchscreen by augmenting the on-screen buttons with appropriate tactile feedback. McAdam *et al.* extended this to large scale table displays showing significant performance improvements with tactile feedback [4]. Poupyrev and Maruyama have examined tactile feedback for interacting with mobile touchscreen widgets [9] and to add physicality to a tablet based computer drawing program [10].

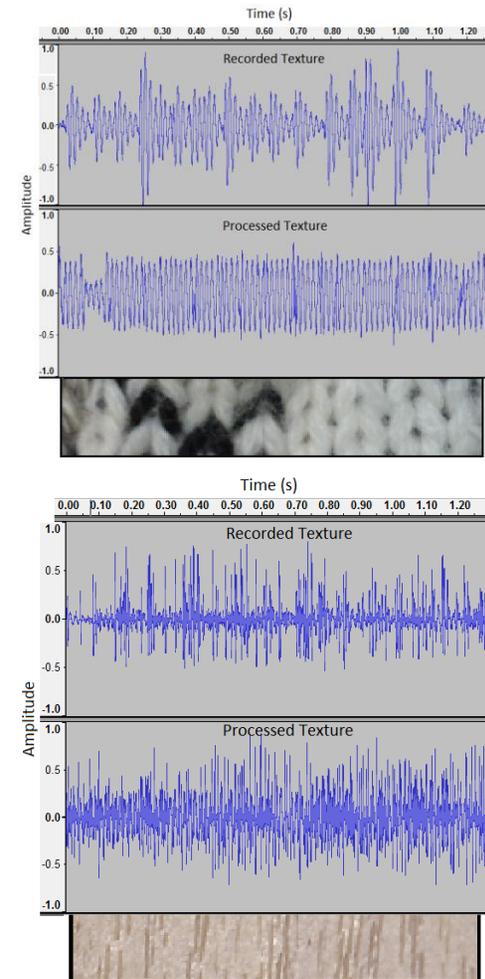
The above examples demonstrate how tactile feedback can augment vision when using a touchscreen device. There is little work however on non-visual touchscreen interaction. Notable exceptions include work by McGookin *et al.* [5]. They note the difficulties that touchscreens present when considering how visually impaired users interact with mobile devices. They demonstrate how the use of a physical raised paper overlay can allow the user to explore the touchscreen non-visually. Similarly, Froehlich *et al.* [1] investigate physical barriers on the screen to assist accessibility for motor impaired users. They demonstrate how these barriers can significantly improve pointing performance on a touchscreen. Strachan *et al.* [12] have demonstrated the use of a physical model-based approach for providing non-visual information to the user of mobile device. They use tactile feedback to convey the sensation of a rotational spring dynamical system with the

goal to employ the user's natural intuition of these systems to provide a greater understanding of the moment-to-moment state of the system. Hall *et al.* [2] investigate T-Bars; a new form of tactile widget that the user activates by dragging across the screen from left to right. Tactile feedback is used to guide and keep the user's finger on the widget.

This paper examines a method of interacting with a touchscreen non-visually by texturing different areas of the screen with vibrotactile textures generated using pre-recorded everyday textures. The aim is to use these textures to allow users to identify and interact with areas of a flat touchscreen non-visually without having to remove the device from their pockets.

### **Texturing a Touchscreen with everyday textures**

Here we support a non-visual browsing mechanism where users can run their fingers over a touchscreen to feel a vibrotactile texture that depends on the functionality of that area of the touchscreen. We employ a physical-based modelling approach to provide textures with a similar feel to everyday surfaces such as wood, wool and paper. An alternative approach would be to use abstract textures. We choose to use model-based textures in this instance as we believe the user will find it easier to map the sensation to the texture. It provides a label for the texture that can be associated with an everyday physical object. Previous work has suggested that abstract textures can be difficult to describe and name [8]. We render our textures using a vibration motor on the back of the mobile device. This work takes inspiration from the research of Yao and Hayward [13]. They demonstrate how a convincing sensation of texture can be produced using signals generated when



**Figure 1.** Two tactile textures generated from the contact microphone being dragged over wool (top), and wood (bottom). Shows the recorded texture, processed loopable texture, and an image of the surface on which the recording was made.

running an accelerometer over a surface. Users ran a probe over a textured surface, feeling the texture remotely in real time through a recoil actuator held in the other hand. Their goal was to enhance a surgeon's perception of surface texture through the surgical tools during a minimally invasive surgical procedure. We use a similar technique to generate pre-recorded textures which are then processed as described below to produce compelling loopable textures for touchscreen interaction.

#### *Generating the Textures*

We captured texture data by running a piezo contact microphone attached to a stiff plastic stylus across various test materials at a constant speed (examples shown in Figure 1). These signals were captured with a standard soundcard. To render textures in response to the onscreen movements, we need to be able to modulate the amplitude and frequency of the texture, so that the tactile signal generated matches the speed of movement of the finger across the touchscreen.

To do this without noticeable discontinuities, the captured signals need to be transformed so that they can be looped and frequency shifted with a minimum of artefacts. We use techniques from audio signal processing. Ideally the input signal would be transformed into a homogenous loop that can be played by starting from any point - this eliminates both clicking at loop points and the artificial sounding repetition effect of retriggering a texture from exactly the same point. We used two approaches to solve this problem: FFT/IFFT based phased-randomization (similar to data-surrogacy techniques used in time-series analysis); and simple cross-fading.

The FFT/IFFT based approach is designed to maintain the spectrum of the original signal but eliminate any temporal modulations; it is equivalent to modelling the signal as coloured noise. The FFT of the signal is obtained, the phase components are randomized between  $-n$  and  $n$ , but the magnitudes are retained, and the IFFT of this signal is obtained. This new signal loops perfectly, without clicking at any point. For some surfaces, which are very homogeneous, this is a reasonable approach. Other surfaces are not well captured by simple frequency spectrum and this technique eliminates some of the key identifiable attributes. The second approach is simply to crossfade the last 25% of the file with the initial 25%. This blends the start and end of the signal together so that the loop sounds and feels clean. As long as the texture does not contain strong regular components (e.g. regular pulsing), the cross-fading effect is barely noticeable.

In both cases, processing is performed offline and we generate a set of texture files that can then be used in an interface and will provide a close approximation of the real texture. Playback is always started from a random point within the file, which also reduces the artificial feel of the texture. Using multiple recordings of the same texture and randomly choosing between them also increases the realism.

#### *Presenting the texture to the user*

This processed texture data is stored as a wav file, which now provides a one dimensional representation of the texture. We then map this to the two dimensional screen using the speed of the user's finger across the screen. On a finger down event we load the appropriate audio file and pause at a random position within the file. Once the user moves above a certain thresh-

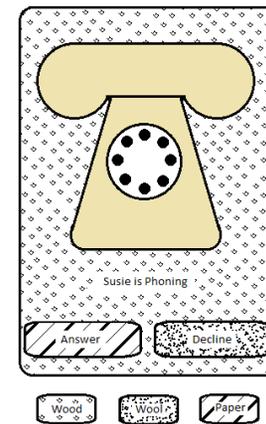
old, the audio stream is unpaused. The rate and volume of playback are altered depending of the rate of movement of the user. To present the texture to a user, we use an EAI C2 Tactor ([www.eaiinfo.com](http://www.eaiinfo.com)) attached to the back of an HTC Hero mobile phone and connected via the headphone socket (as shown in Figure 2). The C2 is a small, high quality linear vibrotactile actuator, which was designed specifically to provide a lightweight equivalent to large laboratory-based linear actuators. The contactor in the C2 is the moving mass itself, which is mounted above the housing and pre-loaded against the skin. This is a higher performance device than the vibration motors commonly found in mobile phones and allows us to explore richer textures in this early phase of our work and to find the key aspects to differentiate textures. We will eventually use an inbuilt actuator.



**Figure 2.** The equipment used. The HTC Hero phone with an EAI C2 Tactor vibrotactile device attached to the back, connected via the headphone socket.

The goal is to create a surface which the user can explore non-visually. Different types of interface controls

can be assigned different textures. Users can then browse the screen by dragging their fingers over the surface until they identify the texture associated with the functionality they seek and then tap to select. Figure 3 demonstrates how this might be used in practice. In this example the user can accept or cancel a call without having to look at the screen. Users can locate and identify the buttons on the screen by their given textures and once found can select using a tap. Alternatively, textures can be associated with the functionality on the screen such as scrolling and panning where the scroll operation would have a different texture associated with it than when the user is browsing the screen with their finger to alert the user to the current state of the system. This technique would be particularly suitable for use with the T-Bar widgets suggested by Hall *et al.* [2] where selection are made by horizontal dragging and different T-Bars could be assigned different distinct textures.



**Figure 3.** An example of how the texturing would be used in practice to accept or decline a call non-visually.

## Conclusions and Future Work

Here we have described a technique for generating compelling vibrotactile textures based on everyday surfaces that can be used to augment touchscreen interfaces. By varying the rate and amplitude of playback with the user's finger speed, we can generate stimuli that feel more like textures than tactile effects.

The next stage of this work will be to demonstrate how this technique performs in a real life setting. This will involve evaluating the performance of users in finding and selecting interface objects using both textures generated as described above and abstract textures. This approach also naturally lends itself to multimodal feedback with the audio files being generated providing an equivalent audio texture. We will further extend the tactile texturing to examine the potential benefits of audio for similar purposes. By exploiting users' inherent knowledge of everyday textures, we hope to provide stimuli that are familiar and memorable. These techniques provide a mechanism for increasing the accessibility of touchscreen devices and allowing non-visual exploration of the screen.

## Acknowledgements

This work was funded by EPSRC grants EP/F023405 and EP/E042740 and Nokia.

## References

- [1] Froehlich, J., Wobbrock, J. O., and Kane, S. K. 2007. Barrier pointing: using physical edges to assist target acquisition on mobile device touch screens. In Proceedings of ASSETS 2007. ACM, pp 19-26.
- [2] Hall, M., Hoggan, E. and Brewster, S.A. T-Bars: Towards Tactile User Interfaces for Mobile Touchscreens. In Proceedings of MobileHCI 2008 (Amsterdam, Holland), ACM Press, pp 411-414
- [3] Hoggan, E., Brewster, S. A., and Johnston, J. 2008. Investigating the effectiveness of tactile feedback for mobile touchscreens. In Proceeding of ACM CHI, 2008. ACM, New York, NY, pp 1573-1582.
- [4] McAdam, C. and Brewster, S.A. Distal Tactile Feedback for Text Entry on Tabletop Computers. in Proceedings of BCS HCI, 2009 (Cambridge, UK).
- [5] McGookin, D. and Brewster, S.A., Jiang, W. Investigating Touchscreen Accessibility for People with Visual Impairments. In Proceedings of NordiCHI 2008. ACM Press, pp 298-307.
- [6] MacKenzie, I. S., Zhang, S. X. Soukoreff, R. W. Text entry using soft keyboards, Behaviour & Information Technology, 1999, VOL. 18, NO. 4, pp 235-244.
- [7] Nashel, A. and Razzaque, S. 2003. Tactile virtual buttons for mobile devices. In CHI '03 Extended Abstracts of ACM CHI, 2003. ACM, pp 854-855.
- [8] O'Sullivan, C. and Chang, A. An Activity Classification for Vibrotactile Phenomena, in Proceedings of Haptic and Auditory Interaction Design, LNCS 4129, 2008, pp 145-156.
- [9] Poupyrev, I. and S. Maruyama. Tactile interfaces for small touch screens. Proceedings of UIST. 2003: ACM: pp 217-220
- [10] Poupyrev, I. and S. Maruyama. Drawing With Feeling: Designing Tactile Display for Pen. Proceedings of SIGGRAPH'2002, Technical Sketch. 2002: ACM: pp. 173
- [11] Ronkainen, S., Häkkinen, J. and Pasanen, L. Effect of Aesthetics on Audio-Enhanced Graphical Buttons. In Proceedings of the International Conference on Auditory Display (ICAD), Limerick, Ireland, July 6-9, 2005.
- [12] Strachan, S., Lefebvre, G., and Zipp-Rouzier, S. 2009. overView: physically-based vibrotactile feedback for temporal information browsing. In Proceedings of Mobile HCI, 2009. ACM, pp 1-4.
- [13] Yao, H., Hayward, V. A Tactile Enhancement Instrument for Minimally Invasive Surgery, Computer Aided Surgery, Vol. 10, No. 4, pp 233-239.