Vote-O-Graph: A Dishonest Touchscreen Voting System

Andrea L. Mascher
The University of Iowa
Iowa City, IA 52242 USA
amascher@cs.uiowa.edu

Paul T. Cotton
The University of Iowa
Iowa City, IA 52242 USA
pcotton@cs.uiowa.edu

Douglas W. Jones
The University of Iowa
Iowa City, IA 52242 USA
jones@cs.uiowa.edu

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract
We present Vote-O-Graph, an experimental touchscreen voting system designed to simulate reported interface issues in existing electronic voting systems. Touchscreen miscalibration and the application of relative touch coordinates in anonymity-preserving user interface event logs are discussed.

Keywords
Electronic voting, touchscreens, miscalibration, event logs

ACM Classification Keywords
H.5.2 User Interfaces: Input devices and strategies

General Terms
Human Factors, Security

Introduction
Electronic voting machines have become prevalent in the wake of the 2000 US presidential election. Such systems are preferred because they prevent overvotes (selecting too many candidates in a given contest), have the potential to reduce undervotes (selecting too few candidates), and provide improved access through multilingual and multimodal interfaces.

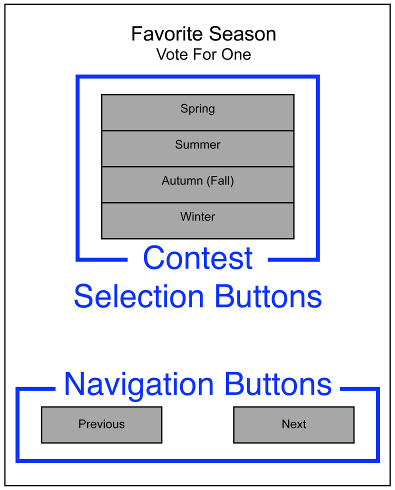

Figure 1. Layout of a Vote-O-Graph contest page

Most concerns regarding electronic voting systems have focused on their security vulnerabilities and lack of verified audit logs, but the 2006 Sarasota County, Florida Congressional District 13 election ("CD13") has brought increased scrutiny of user interface issues with touchscreen voting systems. 14.8% of votes cast on touchscreen systems had undervotes in the CD13 contest, which was several time higher than comparable up-ticket contests such as Senate, Governor, and Attorney General (1.14%, 1.28%, 4.36%, respectively) and more than five times greater than paper ballots (2.5%) [2]. Post-election investigations have proposed that this abnormally high undervote rate was due to user interface issues, namely poor ballot design, and touchscreen miscalibration or insensitivity[1, 2], but the existing event logs for CD13 did not record sufficient information to prove this one way or another.

Existing methods to detect and respond to user interface issues, such as touchscreen miscalibration, record too much information to be used in voting system event logs. The right to a secret ballot is compromised when it is possible to reconstruct how a person voted from the event log. This balance, between the need to protect voter anonymity and the desire to collect the maximum amount of meaningful data for post-election investigations, has prompted several questions:

- What user interface issues can be detected while maintaining voter privacy?
- How can said issues be detected without revealing a voter's selections?
- Are the data for different types of issues differentiable from one another?

We have developed a touchscreen voting system, Vote-O-Graph, to be a testbed for experiments intended to answer these questions. The study described in this paper investigates what anonymity-protecting user interaction data can be maintained in a voting system event log and what measurable differences in behavior exist under a variety of interface issues.

Design

Vote-O-Graph is not designed to be an honest voting machine in the traditional sense. Instead, it is designed to simulate commonly reported touchscreen interface issues. Controlled modifications have been applied to the ballot presentation, calibration, sensitivity and summary screen honesty. Vote-O-Graph also records and maintains anonymity-preserving logs of all user touch events.

System

Vote-O-Graph is a 1500 line Java/Swing application designed to work on any touchscreen notebook computer. Our studies were conducted on a HP tx2510 laptop/tablet running Ubuntu Linux 8.10. This computer has a 12.1"(307mm) screen running at 1200x800 pixel resolution and was configured as a tablet computer in all experiments. The ballot is specified as an XML file.

The visual design of Vote-O-Graph is based on layouts used in existing commercial and experimental voting systems, such as Pvote [5]. Contests are normally presented 1 per page with contest description at the

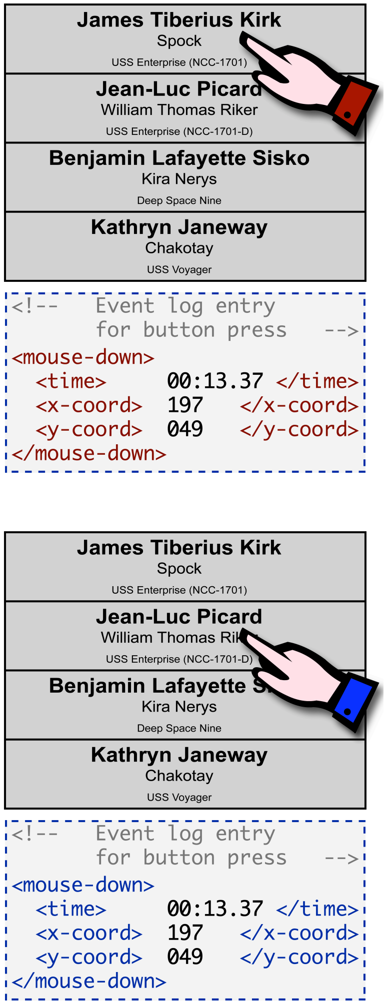

Figure 2. Voter 1 (top) selects Kirk, at the same time that Voter 2 (bottom) selects Picard. Both voters' event logs record the same relative coordinates

top of the screen, candidate selection options presented as a column of adjacent buttons in the middle of the screen. The "Next" and "Previous" navigation buttons are in the lower right and left hand corners, 20 pixels (4.1 mm) from the bottom of the screen. All buttons had a height of 90 pixels (18.4 mm). (See Figure 1.)

Input Event Logs

Logging user interface events is often at odds with the need to protect voter privacy and anonymity. A standard record of a touch event includes a timestamp and absolute touch location with respect to the screen as a whole. These data can be enough information to reconstruct what selections have been made and which voter made them. To prevent this, records of touch events are modified before they are added to Vote-O-Graph's event log.

We record two types of locations that a voter could touch: a button or the background. A touch on the background does not change the state of the ballot or screen, but an excessive number of background touches may indicate a system issue. It is often the case that a background touch is a miss on a nearby button, so to preserve voter privacy, we only record when a background touch occurs, not where.

When a button is touched, we do not record the exact identity of the button. Instead, we record the button type (either contest selection or navigation) and action performed on the button (select, deselect, etc.). The location where the button was touched is recorded as an (x,y) pair relative to the button itself, not to the screen as a whole. This prevents leaking a voter's selection, since touching the same location of the any button would be recorded the same. For example, Figure 2 shows the relative touch coordinates for both "Kirk" or "Picard" recorded as (197,49) even though their absolute coordinates differ. This use of relative touch coordinates allows Vote-O-Graph to anonymize selections the voter made.

User Study

Participants and Environment

To simulate the election experience as closely as possible, we ran studies in locations that are, or resemble, actual voting sites. Participants were recruited from passers-by at our study locations in Johnson County, Iowa.

As of publication, 80 participants have completed the study. The age range was 18-75+ years; 45 were female and 35 were male. Computer and internet experience ranged from none to more than 40 hours a week. 24% of subjects had previous experience with a touchscreen voting system.

Procedure

Participants were told that the study was about "how people interact with voting machines," with no further description of the nature of the study. Participants were instructed to vote any way they wished and encouraged to use the system as they normally would in an election setting.

Participants were free to ask questions if system occurred, but whenever possible we gave minimal information without looking at or touching the system. After voting, participants were given the opportunity to comment on the system.

Task

We conducted randomized, double-blind voting sessions with one of the simulated interface issues described in Table 1. Participants voted on the November 2008 General Election ballot used in Johnson County, Iowa. Given space constraints, only results from miscalibration experiments will be discussed. Additional preliminary findings can be found in [3].

Experimental Group	#
Control: 1 contest per screen, no intentional problems	12
Compressed: Multiple contests on some screens	9
Dishonest: Presidential selection "flipped" on summary screen	14
Insensitive: Delayed response of touch events to simulate insensitivity	20
Miscalibrated: Offset vertical touch coordinates to simulate touchscreen miscalibration	25

Table 1. Experimental groups and subject counts

Touchscreen Miscalibration

Touchscreen devices consist of two completely separate components: a display screen, and the touch input device that overlays the screen. Because of this separation, there is no intrinsic relationship between a point on the display screen and touch sensor directly above it and these components can become out of sync (or "miscalibrated").

If a touchscreen device is miscalibrated by a constant displacement, then all recorded touch coordinates will be offset by the same constant. The offset will be the same, regardless if the coordinate is relative to the screen as a whole or to a target, such as a button, on the screen.

Moffatt discovered that there is a general trend for users to tap below the middle of a target with 82% of target selection errors occurring on the item immediately beneath. Additionally, a target selected in the top 10% of its height is 11 times more likely to be intended for the item above it than for the selected item itself [4]. From this, we hypothesized that vertical miscalibration would impact the average relative vertical coordinate for button presses.

We simulated touchscreen miscalibration by intercepting touch events, perturbing the coordinates by a constant vertical offset. The buttons used in all tests had a height of 90 pixels (18.4mm). Offsets were ±(15%, 25%, 30%) of button height, resulting in physical offsets of ±(2.6, 4.5, 5.5)mm.

Results

As of publication, 4369 vertical touch coordinates have been recorded. 2826 touches were not perturbed. 552 touches were perturbed upwards (the recorded touch was above the physical touch). 991 touches were perturbed downwards (the recorded touch was below the physical touch).

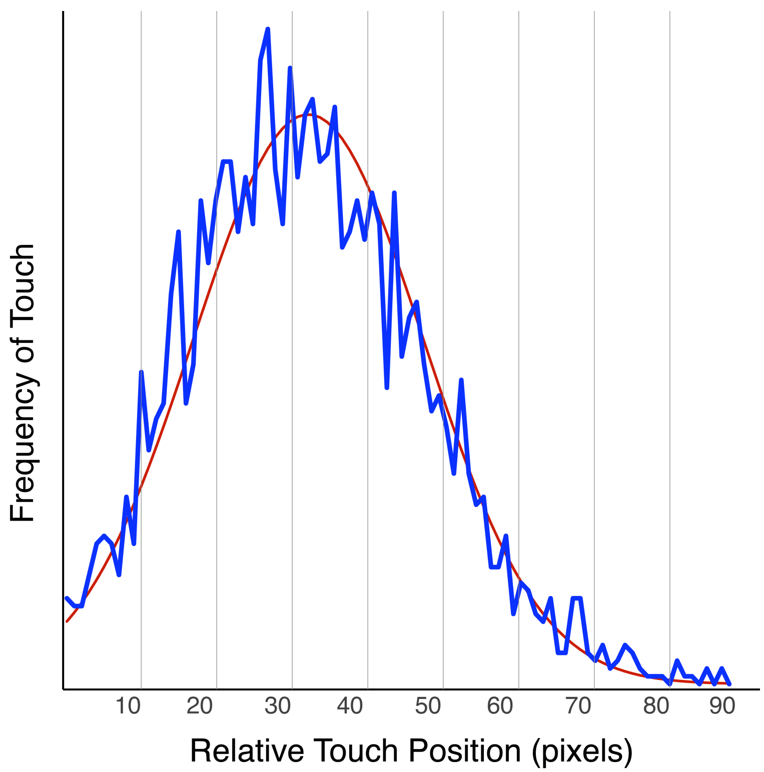

Figure 3. Relative touch positions of all normally calibrated touch events

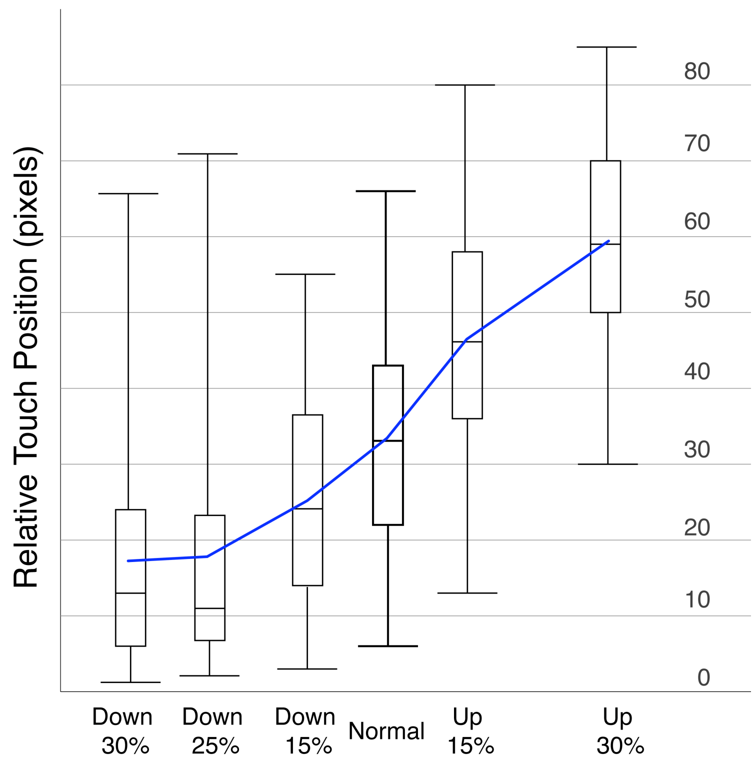

Figure 4. Boxplot of relative touch positions for all offsets. Error bars represent the inner-95% range of coordinates

The average vertical coordinate for normally calibrated touches was approximately 1/3 above the bottom of the button (height=32.44 pixels (6.63mm), sd=15.43 pixels (3.15mm)). (See Figure 3.)

Perturbations in average coordinates for the various miscalibration experiments were proportional to the direction and magnitude of their offsets ($F_{5,2820}=358.6$, $p<0.001$). (See Figure 4.)

Discussion and Future Work

The results of this experiment demonstrate the potential of relative touch coordinates as an anonymity-preserving technique to detect and diagnose touchscreen miscalibration.

Our data agree with Moffatt's findings on the distribution of touches. The tendency to touch targets below the middle was especially pronounced: 2221 of the 2826 (78.6%) unperturbed touches were in the lower half of a button, while only 9 (0.3%) touches were in the top 10%. Perturbed touch coordinates followed similar patterns when readjusted by their initial offset vectors. This consistency in physical touch behavior means that miscalibration that is small with respect to the screen as a whole is still detectable.

Furthermore, records of relative touch coordinates can be used for retrospective analysis or as a real-time diagnostic tool in both voting systems and other touchscreen devices. Consider a touchscreen ATM that has become miscalibrated between maintenance visits. Regular tabulation and transmission of relative coordinates would flag miscalibration without disclosing customer PIN codes.

This work presently contains only a preliminary analysis of data since we are still actively collecting data. As we increase the sample size for the use study we will expand the analysis of this and other hypotheses and control for demographics factors such as age and computer experience.

Acknowledgements

This work was supported, in part, by NSF Grant CNS-05243.

Citations

[1] D. L. Dill and D. S. Wallach, "Stones unturned: Gaps in the investigation of Sarasota's disputed congressional election," Tech. Rep., 2007. Available: http://www.cs.rice.edu/ dwallach/pub/sarasota07.html

[2] L. Frisina, M. C. Herron, J. Honaker, and J. B. Lewis, "Ballot formats, touchscreens, and undervotes: A study of the 2006 midterm elections in Florida," Election Law Journal, vol. 7, no. 1, pp. 25–47, March 2008.

[3] A. L. Mascher, P. T. Cotton, and D. W. Jones "Improving Voting System Event Logs" in Proceedings of the 1st International Workshop on Requirements Engineering for E-voting Systems, 2009.

[4] K. A. Moffatt and J. McGrenere, "Slipping and drifting: using older users to uncover pen-based target acquisition difficulties," in Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility, 2007.

[5] K. P. Yee, "Prerendered user interfaces for higher-assurance electronic voting," in Proceedings of the 2006 USENIX/Accurate Electronic Voting Technology Workshop, 2006.