Towards Publishable Event Logs That Reveal Touchscreen Faults

Andrea L. Mascher Paul T. Cotton Douglas W. Jones

Department of Computer Science
The University of Iowa
Iowa City, Iowa
{*amascher, pcotton, jones*}*@cs.uiowa.edu*

Abstract

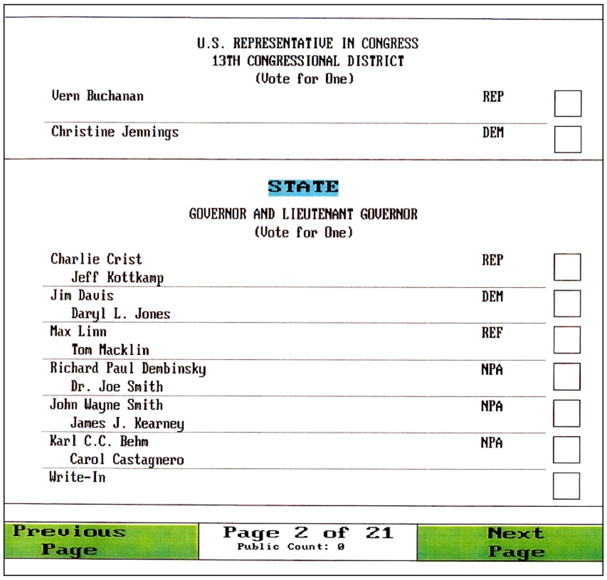
Federal standards require that electronic voting machines log information about the voting system behavior to support post-election audits and investigations. Our study examines interface issues commonly reported in touchscreen voting systems (miscalibration, insensitivity, etc.) and the voter interaction data that can be collected to allow investigation of these issues while at the same time preserving the right to a secret ballot. We also provide empirically derived metrics that can detect these issues by analyzing these data.

1 Introduction

Electronic voting machines have become prevalent in the wake of the 2000 US Presidential election. Such systems have replaced mechanical and punchcard ballots because they prevent overvotes (the selection of too many candidates in a given contest), have the potential to reduce undervotes (the lack of a selection in a contest), and provide improved accessibility through multilingual and multimodal interfaces.

The Help America Vote Act mandated “at least one direct recording electronic voting system or other voting system equipped for individuals with disabilities at each polling place,” and authorized up to $3.9 billion for implementation of its reforms [1]. Despite the recent ubiquity of these systems in American elections, there are still widespread problems with existing voting machines.

Most concerns regarding electronic voting systems have focused on their security vulnerabilities and lack of verified audit logs, but the 2006 Florida Congressional District 13 election (“CD13”) in Sarasota County has brought increased scrutiny to the user interfaces of touchscreen voting systems. 14.8% of votes cast on the ES&S iVotronic touchscreen systems used in that election had undervotes in the CD13 contest, a rate several times higher than for comparable up-ticket contests such as Senate, Governor, or Attorney General (1.14%, 1.28%, 4.36%, respectively) and more than five times greater than the undervote rate for paper ballots (2.5%) used in the same election [10]. 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 [6, 10, 21], but the existing event logs for CD13 did not record sufficient information to test these hypotheses.



U.S. REPRESENTATIVE IN CONGRESS
13TH CONGRESSIONAL DISTRICT
(Vote for One)

Vern Buchanan	REP	□
Christine Jennings	DEM	□

STATE

GOVERNOR AND LIEUTENANT GOVERNOR
(Vote for One)

Charlie Crist Jeff Kottkamp	REP	□
Jim Davis Daryl L. Jones	DEM	□
Max Linn Tom Macklin	REF	□
Richard Paul Dembinsky Dr. Joe Smith	NPA	□
John Wayne Smith James J. Kearney	NPA	□
Karl C.C. Behm Carol Castagnero	NPA	□
Write-In		□

Page 2 of 21
Public Count: 0

Figure 1: Screenshot of Florida’s 2006 Sarasota CD13 and Gubernatorial contests. Image edited to fit column width.

All electronic voting systems are required to maintain an audit trail, more properly called an event log. The requirement for event logs in voting systems dates back to the original voting system standards promulgated by the Federal Election Commission in 1990:

> "All systems shall include capabilities of recording and reporting the date and time of normal and abnormal events, and of maintaining a permanent record of audit information that cannot be turned off. For all systems, provisions shall be made to detect and record significant events (e.g., casting a ballot, error conditions which cannot be disposed of by the system itself, time-dependent or programmed events which occur without the intervention of the voter or a polling place operator)" [8]

Subsequent federal standards have continued to support this requirement [9, 20]. Currently deployed systems rarely record events beyond the minimum listed in the 1990 FEC standard. While these events are useful for a post-election investigation, they are far from sufficient. In many voting system event logs, the only voter interaction recorded is the casting of a ballot. This lack of information recorded in existing event logs hinders investigations into many reported problems, specifically those related to voter experience and intent.

A log of all voter actions should allow easy diagnosis user interface issues, such as touchscreen miscalibration, but this records too much information. 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 ballot secrecy and the desire to collect the maximum amount of meaningful data for post-election investigations, has prompted several questions:

- Which user interface problems can be detected by logging events without revealing voter selections?
- Can different types of problems be differentiated from these event logs?

We have developed a touchscreen voting system, Vote-O-Graph, to be a testbed for experiments intended to answer these questions. The user study described in this paper investigates what user interaction data can be maintained in a voting system event log without threatening ballot secrecy and what measurable differences in behavior exist under a variety of interface issues.

2 Related Work

In many systems that record event logs, the entire history of the system is captured. The level of detail in these logs are sufficient that, given an initial state of the system and the information maintained in the event logs, the final state of the system can be reconstructed. In financial event logs, for example, events typically indicate the amount of money transferred, the source account and the destination account, as well as who authorized the transfer and why. An equivalent event log for a voting system would indicate, at the moment each vote was cast, who cast that vote and for what candidate. Recording such an event log poses obvious threats to the right to a secret ballot. It was observed as early as 1893 that even a sequential record of the votes cast, with no time stamps, is sufficient to allow an observer to determine who cast that vote [17].

Cordero and Wagner proposed using replayable audit logs to create a visual record all of the events in each voting session. By recording touchscreen touches and output events for each voting session, they allow reconstruction of that session in sufficient detail that human-factors problems during voter interaction with the system can be studied in detail [5].

In an effort to anonymize the data, Cordero and Wagner does not store time stamps in the log, and while the sequence of events in each voting session is stored in order, a history independent data structure is needed to store the logs for each voter, so that, after the polls close, it is difficult to tie individual voters to the records of their voting sessions. The lack of time stamps and the use of a history independent data structure mean that Cordero and Wagner's replayable logs must be stored separately from the conventional event log required by current voting system standards.

Despite the efforts to anonymize voting sessions, voters can easily add personal signatures to their replayable event logs. Consider, for example, a voter who has agreed to sell her vote. The vote buyer and vote seller would agree on a pre-determined ballot signature, such as touching each corner of the screen in some prearranged sequence. The vote buyer could examine the replayable log to look for the signature and verify that the vote seller cast a ballot with the agreed selections. Because a ballot signature can be associated with a voter's candidate selections, public release of such a replayable log is problematic. We believe that event logs that cannot be released for public examinations are themselves problematic, so we have sought an alternative.

3 System

Our experimental touchscreen voting system, 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 impact the ballot layout, perceived touchscreen calibration, perceived touchscreen sensitivity, and summary screen honesty.

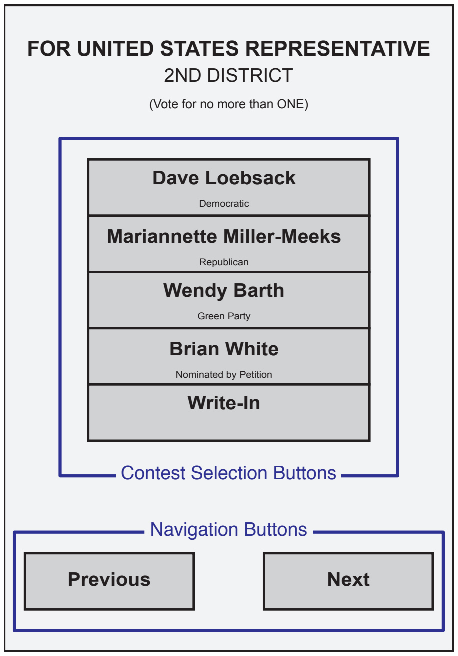


Figure 2: Layout of a Vote-O-Graph contest page. The contest shown is from the November 2008 US House election in Iowa's 2nd district.

Vote-O-Graph is a 1,500 line Java/Swing application designed to work on any touchscreen notebook computer. Our user studies were conducted on a HP tx2510 laptop/tablet running Ubuntu Linux 8.10. This computer has a 12.1" (307 mm) screen running at 1200×800 pixel resolution and was configured as a tablet computer in all experiments. The ballot is specified in an XML file.

The visual design of Vote-O-Graph is based on layouts used in existing commercial and experimental voting systems, such as the ES&S iVotronic or Pvote [25]. Contests are normally presented one per page with contest description at the top of the screen and 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. Voters are required to review their selections and may make updates via a series of summary screens before the ballot is cast. On an update page, the "Next" and "Previous" buttons are replaced by a "Return to Summary" button which spans the width of the screen. All buttons had a height of 90 pixels (18.4 mm). An example Vote-O-Graph screen layout is shown in Figure 2.

4 Preserving Anonymity in Event Logs

In US elections, it is crucial to maintain separation between the identity of the voter and the particular selections made in the voter's cast ballot. When these data can be linked, voters become susceptible to coercion and vote selling.

On the other hand, we want to record as much information about the voter's interaction as possible to allow diagnosis of interface problems. To address these competing concerns, we propose logging additional events, while keeping the following goals in mind:

Standards compliant The new events we record contain timestamps and other elements required under current federal standards.

Ease of integration with existing logging systems The new events we record are conventional timestamped event records comparable to the events already being logged on existing voting systems.

Record novel interaction information The new events support detection and ascribe causes of voting system irregularities.

Avoid compromising secret ballot rights As long as vote data can not be inferred from the event log, then the event logs can be released to the public with little or no modifications or redactions.

Unlike Cordero and Wagner, it is not our goal to provide a record or method to recount or verify election results. Rather, our goal is to allow detection of user interface problems. To do this, our interface-based logging system records three types of data: timestamps, button types, and relative locations.

4.1 Timestamps

When the time of an event is recorded in an event log, it is trivial to link it to the voter who was present at that time. For example, an observer at the polling station could keep track of the times and machines used by voters throughout the day. At some later time, the voters on the observer's list could be cross-checked against the entries in the event log.

Given the requirements for timestamped, sequential entries in event logs, there can never be guaranteed anonymity of voters' identities in any system that can be publicly observed. Therefore, to protect ballot secrecy we do not log button identities or absolute touch coordinates.

4.2 Button Type

For each touch event we record the type of button but not the identity. Our event log shows that the voter made a selection, removed a selection, or navigated to another ballot page, but the candidates selected are not recorded. Retaining information about the type of button touched in the event log provides diagnostic information about where in the ballot irregularities occurred. For example, multiple candidate selects followed immediately by deselects on a single ballot page may indicate that the voter had difficulty with the interface. Recording only the button type prevents revealing a voter's selection, although it can reveal when a voter abstains from a specific contest, depending on how a voter navigates through the ballot. Several different approaches to limit this risk are discussed in Section 8.

4.3 Relative Touch Coordinates

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 or interface issue. It may be 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.

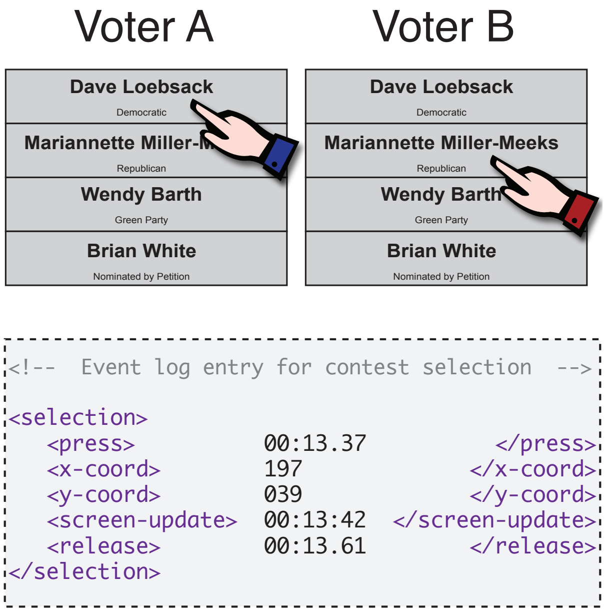
The location where a button was touched is recorded as an (x,y) pair relative to the button itself, not to the screen as a whole. This prevents leaking a voter's selection, since a touch on the same location of any other button would be recorded the same. For example, Figure 3 shows the relative touch coordinates for both "Loebsack" or "Miller-Meeks" recorded as (197,39) even though their absolute coordinates differ. This use of relative touch coordinates allows Vote-O-Graph to record useful information about the voter interaction without revealing the selections the voter made.

5 User Study

5.1 Participants and Environment

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

As of publication, 100 participants have completed the study. The age range was 18–75+ years; 51 were female and 49 were male. Computer and internet experience ranged from none to more than 40 hours a week. 22.5% of subjects had previous experience with a touchscreen voting system.



```
<!--  Event log entry for contest selection  -->

<selection>
   <press>          00:13.37          </press>
   <x-coord>        197             </x-coord>
   <y-coord>        039             </y-coord>
   <screen-update>  00:13:42  </screen-update>
   <release>        00:13.61        </release>
</selection>
```

Figure 3: Relative touch coordinates. Voter A selects Loebsack (upper left), while Voter B selects Miller-Meeks (upper right). The event log entries for both voters are as shown at bottom because both voters' touch events were on the same location relative to their selected buttons.

5.2 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. After demographic data were logged, participants were instructed to vote any way they wished and encouraged to use the system as they normally would in a real election setting. Participants were reminded that their selections would not be recorded.

Participants were free to ask questions, but whenever possible we gave minimal information without looking at or touching the system. After voting, participants were given a questionnaire and notes were taken on any comments made. Voting sessions took 1.33–11.14 minutes (mean=3.78, sd=2.60), depending on the physical and technological abilities of the participant.

5.3 Task

We conducted randomized, double-blind voting sessions with one of the simulated interface issues described in

Group Name	Abbreviation	Description		Number of Subjects
Control	Cont	No intentional problems, one contest per page		13
Compressed Ballot	Comp	Multiple contests per page		11
Dishonest Summary	Dis	Presidential selection changed on summary page		15
Delayed Response	Del-100	Delayed screen response to touch events	100 ms	14
	Del-250		250 ms	20
Touchscreen Miscalibration	Mis Up{amt}	Touch coordinates transformed	Up	11
	Mis Down{amt}		Down	15

Figure 4: User Study Test Groups and Subject Counts

Figure 4. We wanted high levels of recognition for candidates and ballot measures to give the voting act a sense of importance; a ballot with frivolous choices could lead the participants to forget who they voted for when they reviewed the summary screen. Participants voted on the November 2008 General Election ballot used in Johnson County, Iowa [11], but without the option for straight-party voting. The use of a recent election ensures that many contests (especially top-ticket candidates) are still familiar while avoiding the risk that voters might think they have voted in a real election.

The 2008 Johnson County, Iowa ballot had 24 contests, three of which allowed for multiple selections, for a maximum total of 31 selections per ballot. We created two different ballot designs: standard and compressed. The standard ballot placed only one contest per ballot page and was used in the Control, Delayed Response, Dishonest Summary, and Miscalibration experimental groups. The compressed ballot was designed to minimize the number of ballot pages whenever possible and was used to test our hypotheses about banner blindness.

We hypothesize that the events logged by Vote-O-Graph are sufficient to allow the diagnosis of interface problems, but without experimental evaluation we cannot justify requirements that these events be logged by production voting systems. In many cases we expect to be able to diagnose user interface problems by comparing statistical measures of the event logs against norms derived from experiments. These hypotheses are further detailed in Sections 6.1, 6.2, 6.3, and 6.4.

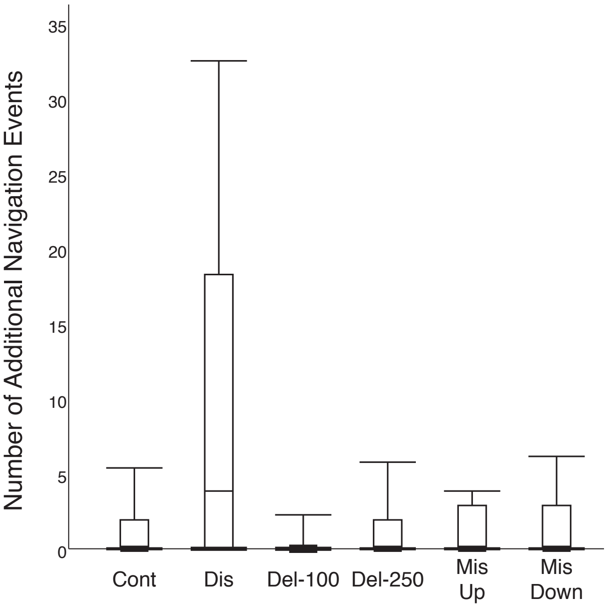


Figure 5: Additional Navigation Events. Whiskers show the inner-90% range, boxes show inner quartile, the dividing line in the box is the median. In this case, the median for all but the Dishonest Summary group was zero.

6 Hypotheses and Results

6.1 Dishonest Summary Screens

Voting machines are complex systems that perform many different functions. As such, they are constructed of multiple layers. A typical voting system consists of firmware that interprets a ballot description that solicits choices through the user interface. Ideally, event logs should be recorded by the lowest system level possible, below all layers that vary from election to election or that are sensitive to candidates or parties.

Ballot designs are especially vulnerable to attack because small changes have the potential not only to mislead and influence voters' selections, but also falsify the record of a vote. For example, a dishonest ballot description could cause a voting machine to record a vote different from the selection shown to voter [22]. With many voting systems dishonesty can be effectively accomplished in the ballot description without changing the firmware. This is often used as a hypothesis to explain the phenomenon that the media and activists have called "vote flipping." Everett's work showed that approximately one out of three voters verify information on the summary page [7]. From this, we expected that about a third of voters would observe a change in their selections on the summary screen and attempt to correct their misrecorded votes via the update page.

To simulate a dishonest ballot description, we changed a subject's initial selection in the Presidential contest on the summary screen. Votes for Barack Obama were switched to John McCain, all other presidential selections (including abstentions) were switched to Barack Obama. Subsequent changes made by the subject to their selection for the Presidential contest were not modified.

Dishonesty in the summary screen led to significant changes in navigation behavior. The standard ballot used in these groups required the subject to navigate forward 37 times to complete the ballot. We expected that a subject discovering a problem at the summary screen would result in an increase in navigation events to update the incorrect selection. Updating each contest requires two navigation events: one to return to the page for a given contest, and another to return the summary screen. We observed this increase in navigation events to update contests and a sharp increase in the number of navigation events back and forth between review pages.

15 subjects experienced a ballot with a dishonest summary screen. 67% of those subjects noticed the dishonesty and reviewed at least one contest. 33% subjects made no reviews and completed the ballot with the minimum number of navigation events as Everett's results predict.

Despite the fact that many subjects apparently did not notice our dishonesty, the effect of dishonesty is still evident in the event logs. On average, subjects with a dishonest ballot reviewed one contest and performed additional navigation between the summary screens resulting in an average of 47.8 navigation events, nearly 12 events more than the minimum of 37. In the Control group, the average number of navigation events was 39.6, only 2.6 higher than the minimum. The results for the Dishonest Summary group and those of all other groups except for Compressed Ballot are presented in Figure 5. This is because the minimum number of navigation events for that ballot layout is different.

More than a quarter of subjects in the Dishonest Summary group performed 50% or more additional navigation events. This result points to a technique which can be used to detect voters responding to abnormal results the summary screen. Setting a threshold based on the number of voters who exceed a certain number of navigation would also be able to detect dishonesty. We recommend more study before determining the optimal settings for such thresholds.

In addition to identifying problems which appear to the voter as a dishonest summary screen, touchscreen miscalibration causes errors which are frequently not corrected until the voter reaches the summary screen. Voters in the Miscalibration groups who corrected contests from the summary screen also perform substantially more navigation events than the Control group. We describe techniques for identifying touchscreen miscalibration in the following section.

6.2 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. When the display screen and touch sensor do not correspond the touchscreen is said to be miscalibrated.

Systems can be deliberately or accidentally miscalibrated by touching the wrong locations during the calibration process, or unintentionally miscalibrated by the voter resting one hand on the screen while voting with the other [12].

If a touchscreen device is miscalibrated by a constant displacement, then all recorded touch coordinates will be offset by the same constant. This offset vector 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 subjects to tap below the middle of a target with 82% of target selection errors occurring on the item immediately beneath the intended target. Likewise, 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 [18]. From this, we hypothesized that vertical miscalibration would impact the average relative vertical coordinate for button presses.

We simulated miscalibration by intercepting touch events and transforming the coordinates by a constant vertical offset vector. The buttons used in all sessions had a height of 90 pixels (18.4 mm). Offsets were ±15%–30% of button height, resulting in physical offsets of ±13–27 pixels (±2.6–5.5 mm).

As of publication, 5,713 vertical touch coordinates have been recorded. 4,069 touches were not perturbed. 653 touches were perturbed upwards (the recorded touch was above the physical touch location). 991 touches were perturbed downwards (the recorded touch was below the physical touch location).

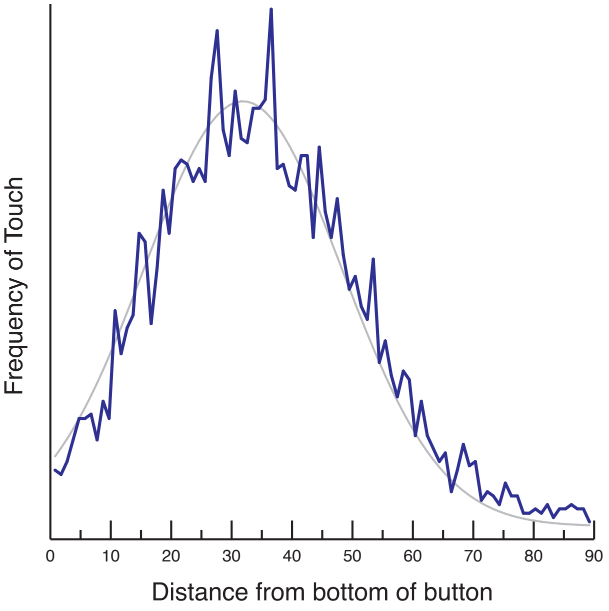


Figure 6: Frequency of relative touch positions on a 90 pixel (18.4 mm) button. Out of 4,069 normally calibrated touch events, 3,091 (76%) of these touches fell below the center of the button, while only 29 (0.71%) of these touches were in the top 10% of the button.

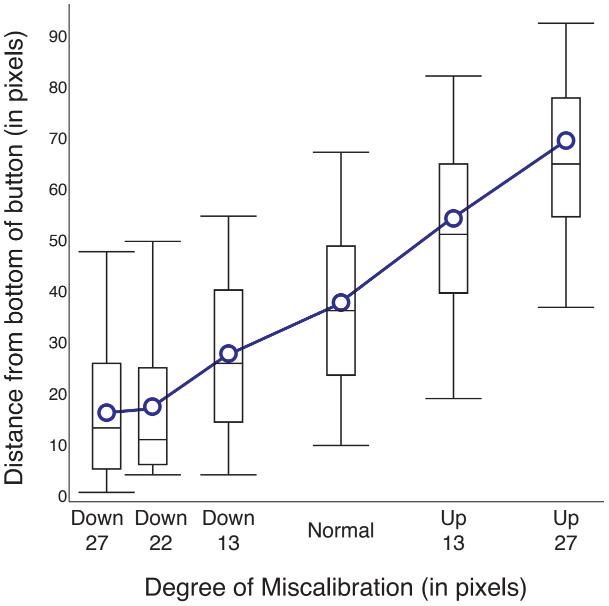


Figure 7: Recorded touch positions on a 90 pixel button for all offset vectors tested. Notation as in Figure 5 with circles marking means. Perturbations in average coordinates for miscalibration vectors were proportional to the direction and magnitude of their offsets ($F_{5,\ 5,707} = 360.19$, $p < 0.001$)

The average vertical coordinate for normally calibrated touches was approximately one third above the bottom of the button (height = 34.28 pixels (7.3 mm), sd = 16.46 pixels (3.5 mm)). (See Figure 6.) Perturbations in average coordinates for the various miscalibration experiments were proportional to the direction and magnitude of their offsets ($F_{5,\ 5,707} = 360.19$, $p < 0.001$). (See Figure 7.)

These results 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: 3,091 of the 4,069 (76%) unperturbed touches were in the lower half of a button, while only 29 (0.71%) touches were in the top 10% of a button. Perturbed touch coordinates followed similar distributions 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.

The consistent distribution of touch positions allows us to use the average location of a touch within a button to measure the degree of miscalibration. A greater upward miscalibration causes a button touch be to be recorded closer to the top of the button. Downward miscalibration causes a touch to be recorded closer to the bottom of the button, as shown in Figure 7. Given the existing tendency to touch the lower half of a button, as the degree of downward miscalibration increases, the median location becomes lower than the mean location, while upward miscalibration maintains a touch frequency distribution similar to that of normally calibrated touches.

Because a running average gives an inaccurate view of the density of the distribution of touch coordinates, we recommend that each relative coordinate be logged to help identify touchscreen miscalibration.

Our results demonstrate that downward miscalibration strongly effects other aspects of interaction with the system. As discussed in Section 6.1, there is a higher incidence of additional navigation events, indicating that selections need to be re-checked more often. Also, the average number of background touches was significantly higher for subjects in Downward Calibration groups (mean = 24.40, sd = 20.08), see Figure 8. Both numbers indicate that subjects are missing their intended targets significantly more when the touchscreen is downward miscalibrated.

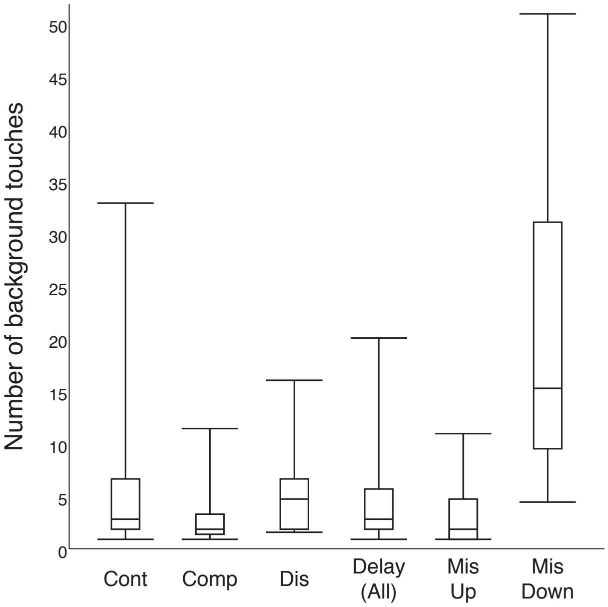


Figure 8: Recorded background touches. Notation as in Figure 5.

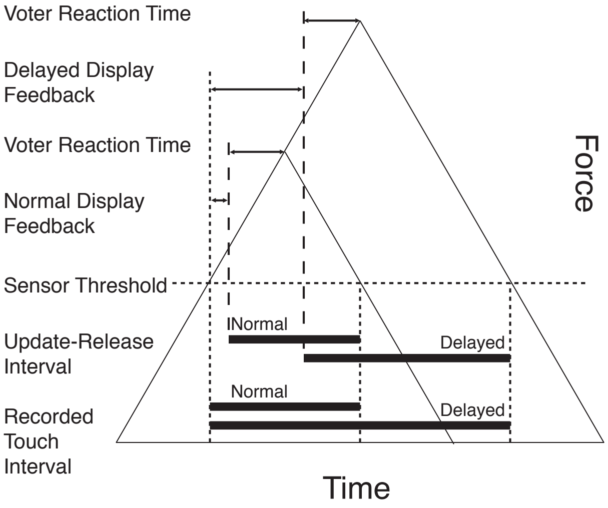


Figure 9: Hypothetical force-delay relationship. The curve of force versus time is not to any scale.

6.3 Compressed Ballots

We experimented with compressed ballots in order to investigate banner blindness. Banner blindness refers to a phenomenon where computer users fail to notice banner ads, even if the ads are prominently placed, large, colorful, or animated [19]. The effect is particularly pronounced if the banners are placed at the top of a screen [2]. It has been suggested that banner blindness may have been at least partly to blame for the unusually large percentage of undervotes in CD13 where the contest was placed at the top of the screen, above a highlighted line [6, 10]. (See Figure 1.)

On our compressed ballots, we placed the US Senate and US House of Representatives contests on the same ballot page. We also compressed the 15 judicial contests down to 6 ballot pages. We expected to see two trends with the compressed ballot style. First, we expected to see a decrease in the rate of votes for the US Senate contest because some voters would miss the contest. Second, we expected to see a slight increase in the rate at which voters change their senatorial votes because the review page would be the first time a voter notices the contest.

Out of the 11 subjects who voted on a compressed ballot only one failed to notice the US Senate contest while voting, but caught the omission on the review screen. Not only was this visible in the event log data, but the voter also commented on the difficulty to find the contest. The low US Senate omission rate may be because our screen layouts and designs were not sufficiently misleading. However, 1 out of 11 (9%) is consistent with the increased undervote rate in the CD13 election. A larger sample size is necessary to be conclusive.

6.4 Touchscreen Insensitivity

Touchscreen insensitivity was reported as one possible cause of the problems in Sarasota CD13 with system vendors acknowledging the existence of delay as intentional [6]. Delay in system response can be quite frustrating and has been shown to markedly increase error rates at 225 ms delay. Shorter, less obvious delays are perceived to be tactile: at 66 ms delay, subjects report that some input devices feel “spongy” [15].

We expected that an increase in delay time would result in greater force being applied to the screen. This effect is illustrated in Figure 9. A number of events occur between the time a subject’s finger touches the screen and releases from the screen. First, as the force between finger and screen crosses the screen’s sensor threshold, the computer is notified of the touch. The Vote-O-Graph program then computes feedback and displays it. The display-feedback time averages 9.6 ms for changes to candidate selection; the median is 5.7 ms. We assume that the subject does not begin releasing until the system response is displayed, and just as the finger pressure increases with time while waiting for a system response, the release is not instantaneous.

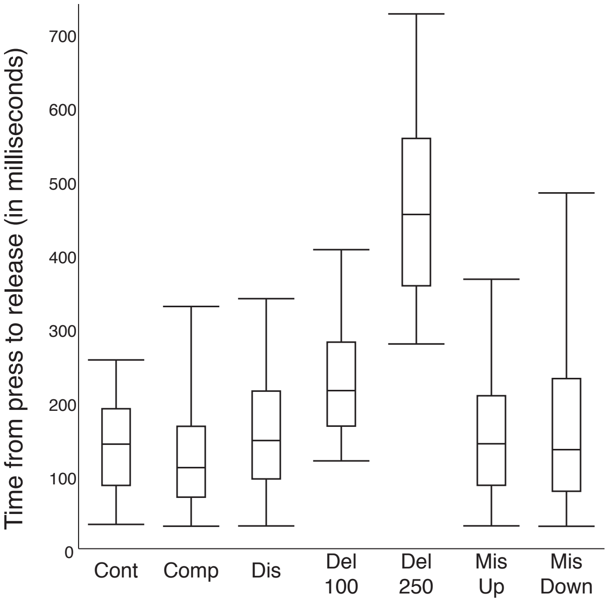


Figure 10: Length of touch times from finger touch to finger release. Notation as in Figure 5.

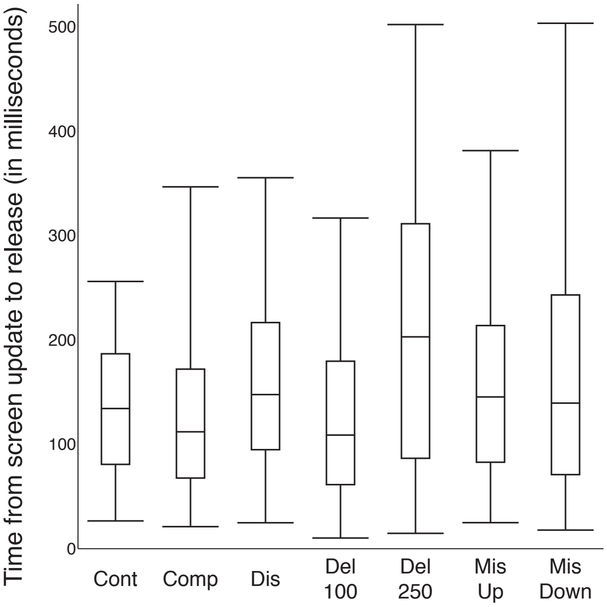


Figure 11: Length of response times from screen update to finger release. Notation as in Figure 5.

We hypothesize that if a voter must press the touchscreen longer or harder to select a ballot item, the update-release interval will increase. To detect touchscreen insensitivity, we record the time feedback is displayed and the finger release times for each candidate selection and de-selection event. We also recorded the actual times the sensor threshold was crossed at touch and release. We did not record this data for screen updates involved with ballot navigation. Our experimental test involved adding a delay of 100 ms or 250 ms to the display-feedback time in order to simulate varying degrees of touchscreen insensitivity. Several subjects who experienced the 250 ms delay commented that they had to press the screen with unexpectedly high force, confirming that a delayed response is indistinguishable from an insensitive touchscreen.

The update-release interval for the combined Control, Compressed Ballot and Dishonest Summary groups averaged 155 ms. The 100 ms Delayed Response group had update-release intervals comparable to this, averaging 124 ms, while the 250 ms Delayed Response group averaged 226 ms. The short to average times for 100 ms group indicate that many subjects did not significantly perceive the added delay, so their behavior did not change. We suspect that this is a result of some subjects not waiting for a screen update before releasing their fingers but instead tapping the screen for a duration of at least 100 ms. Several subjects 250 ms Delayed Response group commented that they had to press the screen with unexpectedly high force, confirming that a delayed response is indistinguishable from an insensitive touchscreen.

7 Unexpected Results

7.1 Insensitivity Deters Proofreading

In the 250 ms Delayed Response group, we observed a marked decrease in the number of subjects who review contests compared with the Control group. Of the 12 subjects in the Control group, 50% subjects made two or more extra navigation events and 17% made six or more extra events. The 100 ms Delayed Response group had behavior similar to that of the Control group: 14 subjects were in the 100 ms Delayed Response group. Less than 50% made any extra navigation events 25% made four or more.

There are two hypotheses to explain the reduction in contest review behavior. Subjects do not bother reviewing contests because the increased delay makes subjects more confident that no review is necessary, or they do not review contests because the increased delay is sufficiently annoying that subjects would rather just get the whole thing over with.

We suspect the latter hypothesis is more likely to be correct. Several subjects in the Delayed Response groups complained about the touchscreen, with more vehement complaints in the 250 ms Delayed Response group. This

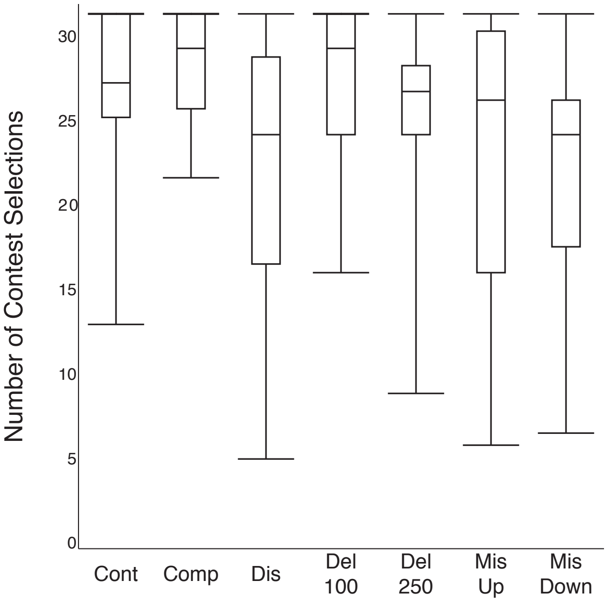


Figure 12: Contest selection rates. Subjects could make a maximum of 31 selections. Notation as in Figure 5.

echos our experience testing the Vote-O-Graph. When the Delayed Response mode was turned on, the touchscreen felt gummy and insensitive so that using it was distasteful. This effect may well have played a role in the Sarasota CD13 contest where the system vendor acknowledged a delayed response with their touchscreen.

7.2 Compressed Ballots can be Good

We observed that subjects who voted on a compressed ballot voted on more contests than any other group (see Figure 12). The increase in selections was primarily in the compressed judicial contest pages.

This increase contradicts the supposition that multiple contests on a single ballot page will increase the residual vote count [13]. During the voting session and in post-voting comments, subjects who did not receive a compressed ballot spontaneously suggested that they would have preferred a compressed judicial ballot.

8 Mitigations

Logging the the number of navigation and candidate selection events for each voter helps identify unexpected results at the summary screen; however this carries risks revealing abstentions from certain contests. This applies to any system that uses a linear navigation model and records both navigation and candidate selection events. If the event log for a voter contains the minimum number of navigation events needed to cast a ballot, then all of the navigation events represent forward navigation, so it is possible to identify which page the voter was viewing at all times. In this context, every abstention will be signaled by two consecutive navigation events with no intervening candidate selection.

An event log that reveals voter abstentions may be acceptable. The right to a secret ballot was instituted to protect voters from coercion, and it is not clear that revealing abstentions subjects voters to the same coercion risk that they would face if their selections were revealed. We take no stand on this public policy question, so in the event that leaking information about abstentions is a problem, we suggest the following mitigation measures.

One way to mask abstentions is to provide an explicit "abstain" option for all contests. Selecting this option would record a selection event identical to other candidate selections. In addition to obscuring voter abstentions, this scheme ensures that there are no unintentional undervotes.

Another option is to record only backward navigation events, since all such events are extra events. The number of backward navigation events with our navigation model is half the difference between the total number of navigation events in a voting session and the minimum number of navigation events per voted ballot. This option still leaks the total number of candidates selected on a ballot, while hiding the contests involved. If we want to guarantee the right of a voter to anonymously cast a completely blank ballot, this option will not suffice.

If we opt not to force explicit selection of an abstention option and we wish to avoid leaking information about which voters cast blank ballots, we must not record candidate selection events. In this case, we can still learn about touchscreen miscalibration from two classes of events: candidate deselection events and background touches. We could also record, for each voting session, the average position within candidate selection and navigation buttons, provided that all buttons are the same dimension. Similarly, we can record, for each session, the average time between displaying feedback and button release.

Note that all of these mitigation options except for explicit abstention eliminate the opportunity to estimate which contest was the source of trouble in a problem ballot. It is not always possible to detect which page is problematic with the primary technique we proposed since we do not differentiate forward and backward navigation. However, in cases where a subject navigates back removes a selection and then makes a new selection it can be inferred that they must have navigated backward, since they could not remove a selection on a contest the first time they see it.

9 Future Work

Further examination of the impact on selection rates using compressed ballots is needed. Several authors have suggested that displaying multiple contests on a screen can increase undervotes [3, 13]. However, some have noted undervote rates increased on longer ballots, suggesting that some voters may become fatigued. Our results suggest that compressing ballots is indeed valuable, but we hesitate to make specific ballot design recommendations without further work.

Some of our experimental groups are too small to allow us to draw firm conclusions. We only had one subject who clearly exhibited banner blindness. It would be useful to enlarge this experimental group significantly.

In testing the impact of touchscreen sensitivity, it would be useful to use a force-sensing screen so that we could directly measure the force versus time behavior during touch events. This would be particularly valuable in a study of the conditions under which some users tap the screen without waiting for a response, and those who touch the screen until they see a response.

We required voters to navigate linearly through the entire ballot before visiting the summary screen which served as a menu for navigating back to contests to correct errors. We do not know how many voters found the change of navigation scheme from linear to menu-based to be confusing. Experiments with other approaches to ballot navigation are clearly needed.

We required explicit deselection of candidates before another candidate could be selected after voters had selected the maximum number of candidates permitted in a contest. We noted that this caused difficulty for some voters attempting to change their selections. There are other models. Consider, for example, first-in-first-out selection where, after a voter selects the maximum permitted number of candidates, additional selections cause deselection of the oldest previous selection. The impact of such alternative models should be explored.

10 Conclusion

Our study demonstrates several types changes to voter behavior under different circumstances which lead us to make new recommendations for both event logs and ballot layout. The increase in votes cast on judicial retention contests in the Compressed Ballot group demonstrates that placing multiple contests on a page can decrease undervotes in some circumstances, particularly when the contests are of the same type.

The changes we observed in voter behavior when using a malfunctioning system can be used to develop a set of decision rules to help identify those malfunctions, which should lead to new requirements for voting system event logs that increase the likelihood a post-election audit could properly identify abnormalities in voting system behavior.

Recording the frequency at which voters navigate back to certain contests from the review screen helps identify contests which were undervoted due to poor ballot design. It could also indicate the presence of a dishonest ballot design. When voters navigate back to contests they have previously visited, this indicates that something is wrong. Touchscreen miscalibration, particularly downward miscalibration, leads to increased backward navigation rate, but a dishonest presentation on the summary screen leads to a far greater effect.

We can distinguish the effect of touchscreen miscalibration from a dishonest voting machine by looking at the background touch rate and changes in the average touch location in the direction of the miscalibration. In addition, the average vertical position of a touch relative to the button touched is a sensitive measure of the quality of touchscreen calibration.

The interval of time between visual feedback from a touch, and the finger release is an effective measure of the sensitivity of the touchscreen.

While we have not proposed specific decision rules for diagnosing problems with touchscreen voting systems, our results support a requirement that voting machine event logs include records of touch duration, location relative to the touched button, background touches (with no location information) and backward navigation events. The changes recommended in this work could significantly strengthen the routine audits of voting systems and provide investigators more tools to diagnose reported problems in elections, while preserving voters' right to a secret ballot.

11 Acknowledgments

This material is based upon work supported by the National Science Foundation under A Center for Correct, Usable, Reliable, Auditable and Transparent Elections (ACCURATE), Grant Number CNS-0524745.

References

[1] 107TH U. S. CONGRESS. Help America Vote Act of 2002, Oct. 2002.

[2] BENWAY, J. P. *Banner Blindness: What Searching Users Notice and Do Not Notice on the World Wide Web.* PhD thesis, Rice University, Houston, TX, USA, Apr. 1999.

[3] COMMISSION ON LAW AND AGING, STANDING COMMITTEE ON ELECTION LAW, AND COMMISSION ON MENTAL AND PHYSICAL DISABILITY LAW. *Report to the House of Delegates — Approved by the ABA House of Delegates on August 13, 2007.* American Bar Association, Aug. 2007.

[4] COMMITTEE ON NATIONAL SECURITY SYSTEMS. National information assurance glossary. Tech. Rep. 4009, June 2006.

[5] CORDERO, A., AND WAGNER, D. Replayable voting machine audit logs. In *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop* (July 2008).

[6] DILL, D. L., AND WALLACH, D. S. Stones unturned: Gaps in the investigation of Sarasota's disputed congressional election. Apr. 2007.

[7] EVERETT, S. P. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection*. PhD thesis, Rice University, Houston, TX, USA, May 2007.

[8] FEDERAL ELECTION COMMISSION. Performance and test standards for punchard, marksense and direct recording electronic voting systems. Tech. rep., Federal Election Commission, Jan. 1990.

[9] FEDERAL ELECTION COMMISSION. Voting systems performance and test standards. Tech. rep., Federal Election Commission, 2002.

[10] FRISINA, L., HERRON, M. C., HONAKER, J., AND LEWIS, J. B. Ballot formats, touchscreens, and undervotes: A study of the 2006 midterm elections in Florida. *Election Law Journal 7*, 1 (Mar. 2008), 25–47.

[11] JOHNSON COUNTY IOWA AUDITOR'S OFFICE. November 4, 2008 Presidential Election, Nov. 2008.

[12] JONES, D. W. Observations and recommendations on pre-election testing in Miami-Dade County. Sept. 2004.

[13] KIMBALL, D. C., AND KROPF, M. Voting technology, ballot measures, and residual votes. *American Politics Research 36*, 4 (2008), 479–509.

[14] KOHNO, T., STUBBLEFIELD, A., RUBIN, A. D., AND WALLACH, D. S. Analysis of an electronic voting system. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy* (May 2004).

[15] MACKENZIE, I. S., AND WARE, C. Lag as a determinant of human performance in interactive systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Apr. 1993), pp. 488–493.

[16] MCCLURE, N. L., WIELAND, R. D., BABBITT, V. L., AND NICHOLS, R. A. Precinct voting system, June 2006.

[17] MCTAMMANY, J. Balloting device, Aug. 1893. US Patent no. 502,743.

[18] MOFFATT, K. A., AND MCGRENERE, J. 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* (Oct. 2007), pp. 11–18.

[19] PAGENDARM, M., AND SCHAUMBURG, H. Why are users banner-blind? The impact of navigation style on the perception of web banners. *Journal of Digital Information 2*, 1 (2001).

[20] U.S. ELECTION ASSISTANCE COMMISSION. Voluntary voting system guidelines. Tech. rep., 2005.

[21] YASINSAC, A., WAGNER, D., BISHOP, M., BAKER, T., DE MEDEIROS, B., TYSON, G., SHAMOS, M., AND BURMESTER, M. Software review and security analysis of the ES&S iVotronic 8.0.1.2 voting machine firmware, final report. Tech. rep., Security and Assurance in Information Technology Laboratory, Florida State University, Feb. 2007.

[22] YEE, K.-P. Prerendered user interfaces for higher-assurance electronic voting. In *Proceedings of the 2006 USENIX/ACCURATE Electronic Voting Technology Workshop* (Aug. 2006).

[23] YEE, K.-P. *Building Reliable Voting Machine Software*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, Dec. 2007.

[24] YEE, K.-P. Extending prerendered-interface voting software to support accessibility and other ballot features. In *Proceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop* (Aug. 2007).

[25] YEE, K.-P. Pvote. http://pvote.org/, Mar. 2007. version 1.0 (beta).