Improving Voting System Event Logs

Andrea L. Mascher, Paul T. Cotton, Douglas W. Jones
Department of Computer Science
The University of Iowa
Iowa City, Iowa, USA
{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 what additional voter interaction information should be collected to allow investigation of human factors issues of the voting systems used in an election, while at the same time preserving voter privacy. We have focused on simulating touch screen interface errors that have been hypothesized as the cause of problems in past elections, such as miscalibration and insensitivity. The preliminary data gathered indicates that event logs which record voter interaction information may allow investigators to detect the existence of interface problems in deployed voting systems. This information can be collected without compromising secret ballot rights. We believe that any voting system using a touch screen interface could benefit by logging these events.**

I. Introduction

Since the passage of the Help America Vote Act, which requires "at least one direct recording electronic voting system or other voting system equipped for individuals with disabilities at each polling place," electronic voting systems have become commonplace in American elections [1]. Despite the ubiquity of these systems, there are still widespread problems with existing voting machines.

All electronic voting systems are required to maintain an audit trail (or more properly, an event log) which allows for the "examination of the sequence of events" [2]. 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)" [3] (emphasis added)

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

Subsequent federal standards continue to support this requirement [4], [5]. Current systems rarely record events beyond the minimum listed in the 1990 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 prevents investigations into many reported problems, specifically those related to voter experience and intent.

The various post-election investigations of the 2006 Sarasota County Congressional District 13 election ("CD13") [6], [7], [8] clearly demonstrated the inadequacy of event logs maintained by the system used in that county. In that race, there was 13% undervote which was several times greater than the Senate, Governor, Attorney General and similar top ticket races (1.14%, 1.28%, 4.36%, respectively). Records of voter complaints include omission of the CD13 race on either the ballot or review screen; touch screen insensitivity; and sluggish response time. Despite extensive post-election investigations, none of the information recorded by the voting system, the ES&S iVotronic, allowed unambiguous substantiation of any of these voter complaints.

Our study examines what events a voting system could record that could allow diagnosis of the cause of errors such as those encountered in the CD13 race. To study these errors we have designed a voting system which can simulate these errors and allow for testing of user behavior under each of several incorrect system behaviors.

II. Related Work

In many systems that record event logs, the logs capture the entire history of the system. The level of detail in these logs are sufficient such that, given the initial state of the system and the information in the event logs, the final state of the system can be computed. 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. As John McTammany pointed out in 1893, even a sequential record of the votes cast, with no time stamps, is sufficient to allow an observer to determine who cast that vote [9].

Cordero and Wagner proposed using replayable audit logs to record all of the events in each voting session. By recording all of the visual output events in each voting session, they allow reconstruction of that session in sufficient detail that human-factors problems during voter interaction can be studied in detail [10].

In an effort to anonymize the data, Cordero and Wagner do not store time stamps in their log, and while they store the sequence of events in each voting session in order, they use a history independent data structure to store the logs for each voter, so that, after the polls close, it is difficult to tie individual voters to 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-arranged ballot signature, such as touching each corner of the screen in clockwise order. 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.

III. User Interface Events

What events can be included in a time-stamped event log that offer a hope of diagnosing user interface problems without threatening the voter's right to a secret ballot? We propose several classes of events here. Records of these events must anonymize voter interaction such that none of the recorded information can reveal the voter's selections. Yet the anonymized logs must retain enough information to be useful in post-election investigations. Currently, all events are recorded in sequential order which makes it possible to identify when a voter abstains from certain races. Possible solutions to this problem are examined in later sections.

A. System Events

System events are actions by the voting system that impact how the voter may interact with the system. For example, a voter cannot make selections when a ballot is not loaded. The log records events in a single chronologically ordered stream with each voter session bracketed by INITIALIZE and CAST events. The time stamped system events we record are:

- INITIALIZE: Load a new ballot. The ballot style is also recorded when multiple ballot styles are supported.
- UPDATE: Report any change to the screen (highlight, unhighlight, change page)
- CAST: Finalize voter session.

Note that we do not record the type of update events but each update follows either a selection, navigation or initialization event, from which type can be inferred.

B. Input Events

We anonymize the record of voter input events by recording the minimum information needed to diagnose problems. 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(See Section IX), so to preserve voter privacy, we only record when a background touch occurs, not where.

When a button is touched, we record the button type (either candidate selection or navigation) and button action (select, deselect, or the navigation destination). The location where the button was touched is recorded as an (x,y) pair of the internal pixel coordinates relative to the upper left hand corner of button itself, not to the screen as a whole. This prevents leaking a voter's selection, since touching the same position of the button for either "Candidate A" or "Candidate B" would be recorded as identical event records. We use these relative coordinates to anonymize which candidate the voter selected. This does not prevent a voter from signing their session in the event log, but such a signature would only allow someone to prove that they voted, not how. We record the navigation destination screen type (Candidate Selection, Write In, Summary, Update) to give diagnostic information about the approximate location in the ballot where issues occur, but we do not record the exact identity of the button or page.

Navigate Button Actions:

- Next/Previous screen
- Write In/Return to Ballot page
- Review/Return to Summary
- Cast Ballot

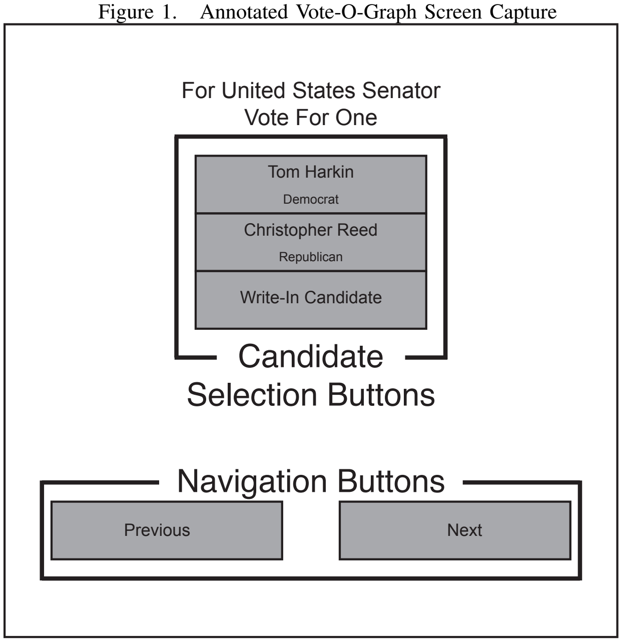
Candidate Button Actions:

- SELECT: Highlight a button.
- DESELECT: Unhighlight a highlighted button.
- I-SELECT: (Invalid Select): The button cannot be highlighted because the maximum number of buttons has already been selected.

IV. System Design

The visual design of the voting system we used in our experiments, which we call Vote-O-Graph, is based on the layout used in Pvote [11]. An example screen capture of the senatorial race for Johnson County, Iowa is shown in Figure 1. Candidate selection buttons can be seen in the column of buttons in the middle of the screen. The 'Next' and 'Previous' navigation buttons can be seen in the lower right and left hand corners. The navigations buttons are separated from the bottom of the screen by 20 pixels(4.1 mm). On an update page, the 'Next' and 'Previous' buttons are replaced by a 'Return to Summary' button which spans the width of the screen.

Figure 1. Annotated Vote-O-Graph Screen Capture



Unlike Pvote, the Vote-O-Graph is not designed to be an honest voting machine in the traditional sense. Instead, it is designed to allow controlled modifications to ballot presentation, calibration, sensitivity and summary screen honesty with logging of user interactions. We decided not to record or tally actual votes so that it would be possible to use ballots from recent elections without a risk of invading the privacy of our study subjects.

Vote-O-Graph is a 1500 line Java/Swing application designed to work on any touch screen notebook computer. Our studies were conducted on a Hewlett-Packard tx2510 touch screen laptop running Ubuntu Linux 8.10. This laptop has a 12.1" (307mm) screen running at 1200x800 resolution. The ballot is specified as an XML file whose size is dependent on the number of contests in an election.

We used the USB interface to connect a pollworker controller. In a production voting system, the pollworker controller would be used to enable the voting machine and select a ballot style for each voter. See, for example, the Judges Booth Controller used with the Hart InterCivic DRE voting system [12]. In Vote-O-Graph, we use the pollworker interface to collect demographic data for each new voter before beginning a new session for that voter. The pollworker controller consists of a small LCD display to prompt the poll worker with questions about voter demographics and a numeric keypad for data entry. Vote-O-Graph currently logs 15 different details about each voter including age, computer and web experience, and voting technologies with which the voter is familiar.

V. User Study

We hypothesize that the events logged by Vote-O-Graph will be sufficient to allow the diagnosis of some 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 VI, VII, VIII and IX.

A. Participants and Environment

Our main experimental priority was to simulate the election experience as closely as possible. Specifically, we wanted to run studies in locations that are, or have attributes that closely resemble, actual voting sites. Characteristics that we looked for include: multi-use facilities, poor acoustics or ambient noise, visual distractions, crowded conditions, less than ideal staging areas (no chairs for waiting, etc.). Locations that we selected include: public libraries, shopping centers, and farmers' markets.

A second priority was to recruit participants that are representative of the average American voting population. Our inclusion criteria was that all participants were eligible voters in Johnson County, Iowa; on average, it should be noted that residents of this county are younger, more liberal, more educated and less ethnically diverse than the general voting population. To address this, we recruited study locations specifically targeted for these under represented populations such as: retirement communities, senior centers, political party gatherings, and multicultural events. We recruited participants from passers-by at our study locations.

As of publication, 59 participants have completed the study. The age range was 18–75+ with a median in the 31–45 year age cohort; 40 were female and 19 were male. Education levels ranged from high school diploma to PhD. Computer and internet experience ranged from none to more than 40 hours a week. Voting sessions took 5 to 15 minutes, depending on the physical and technological abilities of participants.

B. Ballot and Manipulation

Another priority was to create a ballot that resembled commercial voting systems in both layout and content. First, we implemented user functionality on par with existing voting systems. Our ballot presentation included adjacent candidate buttons (i.e., no gutters between selections), a write-in candidate option, and a red-green colorblind friendly palate. The ballot was arranged as a series of screens with a linear navigation structure (Previous Screen and Next Screen buttons) followed by a summary screen from which a voter could either cast the ballot or navigate back to an update page for any race where selections could be changed. We then intentionally introduced several simulated and controlled defects to the touchscreen interface to allow for tests of our hypotheses.

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 initially voted for in the case of dishonest summary. We choose the 2008 General Election ballot for Johnson County, Iowa [13] because it is old enough that ballot items are no longer pending, but recent enough that many contests (especially top-ticket candidates) are still familiar.

The 2008 Johnson County ballot had 24 races, 3 of which allowed for multiple selections, for a maximum total of 31 selections per ballot. We created 2 different ballot designs: standard and compressed. The standard ballot placed only one race per ballot page and was used in the calibration, sensitivity, dishonest summary, and control 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.

C. Task Description

The goal of this study was to explore the hypotheses described below. We conducted randomized, double-blind voting sessions of the types listed in Figure 2. 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 information was logged, participants were asked to vote any way they wished and encouraged to use the system as they normally would. Participants were reminded that the system would not log how or for what they voted. Participants were free to ask questions if system difficulties 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.

Figure 2. Test Types and Subject Counts

Group Name	Description	#
Control	No Intentional Probs. 1 Race per page	10
Dishonest	Summary Changed	11
Compressed	Many races per page	7
Delay 100	100ms delayed response	9
Delay 250	250ms delayed response	7
Calibration +15%	Touches move up 15%	3
Calibration +25%	Touches move up 25%	0
Calibration +30%	Touches move up 30%	1
Calibration +50%	Touches move up 50%	1
Calibration -15%	Touches move down 15%	4
Calibration -25%	Touches move down 25%	2
Calibration -30%	Touches move down 30%	3
Calibration -50%	Touches move down 50%	1

VI. 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 [14]. The effect is particularly pronounced if the banners are placed at the top of a screen [15]. 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 race was placed at the top of the screen, above a highlighted line [6], [7].

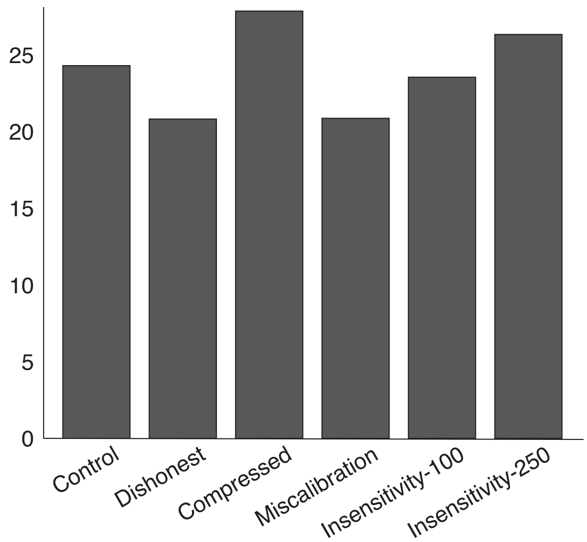
On our compressed ballots, we placed the senatorial and congressional races on the same ballot page. We also compressed the 15 judicial races into 6 ballot pages. We expected to see two trends with the compressed ballot style. First, we expected to see a decrease in the rate of votes for the senatorial race because some voters would miss the race. 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 race.

Out of the 7 subjects who voted on a compressed ballot only one failed to notice the senatorial race while voting, but caught the omission on the review screen. This low senatorial omission rate may be because our screen layouts and designs were not sufficiently misleading. However, 1 out of 7 is consistent with the increased undervote rate in the CD13 election. Our current sample size is too small to be conclusive.

Although it was not a planned outcome, we observed that subjects who voted on a compressed ballot voted on more races than any other group (see Figure 3). The increase in selections was primarily in the compressed judicial race pages.

This seems to contradict the supposition that multiple selection options on a ballot page will increase the residual vote count [16]. In post-voting comments, subjects indicated that they preferred the compressed style of ballot, especially when the races were alike on a given page (i.e., all state Supreme Court or all circuit court). This finding indicates

Figure 3. Average Number of Selections



that multiple race ballot pages are not always bad.

VII. 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.

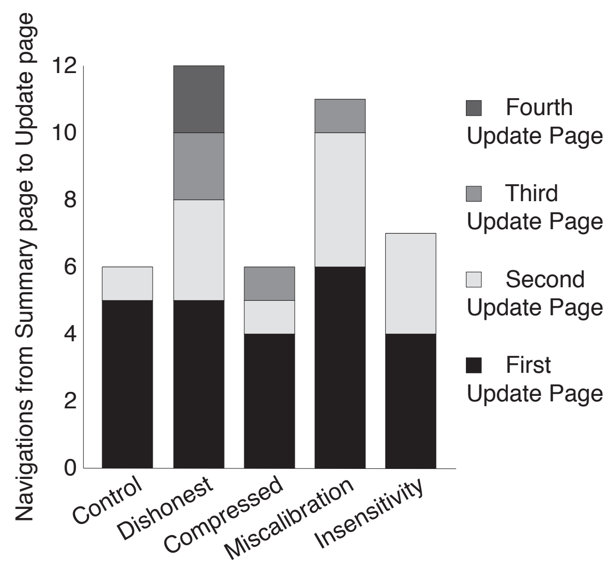
Ballot designs are especially vulnerable to attack because small changes have the potential not only to mislead and influence voters' selections, but also falsify the record of a vote. For example, a dishonest ballot description could cause a voting machine to record a vote different from the selection shown to voter [17]. 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 [18]. 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 Presidential vote on the summary screen as follows: votes for Barack Obama were switched to John McCain, all other presidential selections (including no vote) were switched to Obama. There were no additional switches after a subject reached the summary screen.

Of the 11 subjects who received a dishonest ballot, 5 navigated back to an update page from the summary screen. Approximately the same percentage of subjects in other groups also navigated back. All subjects in the dishonest group who did so visited at least the Presidential race and 60% made additional update page visits. This rate is on par with that observed in the miscalibration and insensitive groups and more than double that for the compressed and control groups.

In the case of the dishonest summary group, out of all update page visits, 58% were from voters who had already visited at least one update page. This rate was slightly greater than that of the miscalibration and insensitive groups and more than triple that of the control group. This relatively high rate indicates that individual subjects in the dishonest group were more likely to look at a larger amount update pages per subject than subjects in other groups.

Figure 4. Navigation to Contest Pages from the Summary Screen



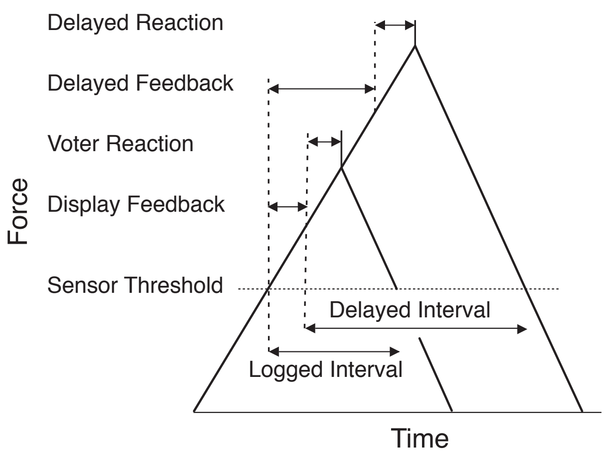
In general, the dishonest summary group has an update screen rate that is similar to that of the miscalibration and insensitive groups. This is not surprising given that one is likely to be more careful and thorough when dealing with a "broken" system. For example, one subject completed 4 separate presidential updates "just to be sure" that the system had not changed the race selection again. These trends indicate that the number of attempts to change selections made may be a meaningful metric to add to the number of voters who change their ballots.

VIII. Touch Screen Insensitivity

Touch screen 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, users report that some input devices feel "spongy" [19].

We expected that an increase in delay time would result in greater force being applied to the screen. This effect is illustrated in Figure 5. Several subjects who experienced the simulated delay touch screen test commented that they had to press the screen with unexpectedly high force, confirming that a simulated delay is indistinguishable from an insensitive touch screen.

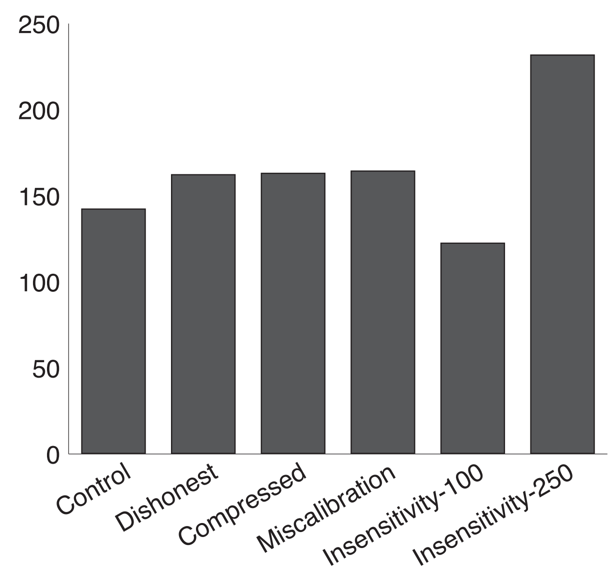
Figure 5. Hypothetical Force-Delay Relationship



We hypothesize that if a voter must press the touch screen longer or harder to select a ballot item, the time between the a change in button highlighting and the voter lifting their finger will increase. To detect touch screen insensitivity, we record the button highlight and release times for each candidate selection and de-selection event because those where the only events which change button highlighting. We also could have monitored delay time by recording the time between a screen update resulting from pressing a navigation button and the lifting of the finger. Our experimental test of this hypothesis delayed the touch screen response time by 100ms or 250ms in order to simulate varying touch screen insensitivity and recorded the times discussed above to allow for an examination of a correlation between touch screen response time and duration of voter touch.

In all experimental groups except the insensitive group, average release times varied from 142.6 to 164.7 ms. Surprisingly, the average release time in the 100ms delay case was slightly shorter than other tests at 122.8ms. We expect that this is a result of the small sample size for this test. However, the 250ms delay test displayed a larger difference: the average response time was 232 milliseconds.

Figure 6. Delay Response Times(ms)



As noted above, we also noticed a decrease in the frequency of voters returning to an update page from the summary screen, suggesting that an insensitive touch screen frustrates voters and reduces review rates.

IX. Touch Screen Miscalibration

Touch screen 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. To make coordinates on the display screen and touch sensor correspond, the software for the touch screen interface must be calibrated. When the display screen and touch sensor do not correspond the touch screen 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, as demonstrated by Jones in 2004. [20].

We simulated miscalibration by intercepting touch events, transforming the coordinates by a constant vertical offset, before checking if those coordinates overlapped with a button. The buttons used in all tests had a height of 90 pixels(18.4mm). Offsets were $\pm(15\%, 25\%, 30\%, 50\%)$ of button height, resulting in physical offsets of $\pm(2.6, 4.5, 5.5, 9.2)$mm.

We refer to an offset as upward when the location the system records as the position of a touch is above the location of the finger on the screen, and downward when the recorded position of the touch is below the location of the finger.

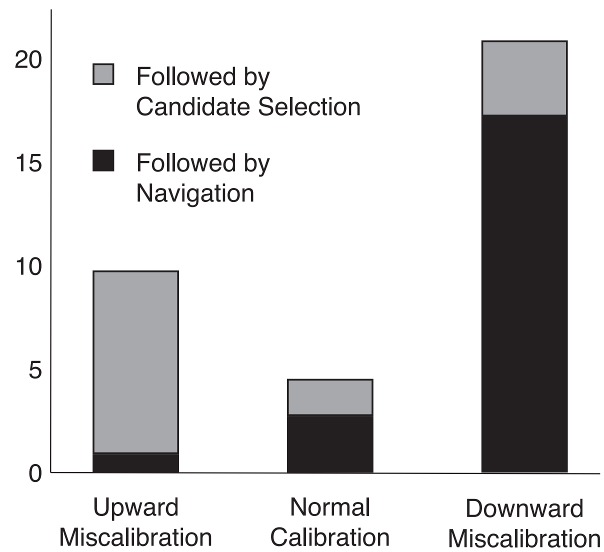
We expected to see two trends in a miscalibrated system: an increase in the number of times voters miss buttons, and an increase in the rate at which voters change their selections, because their touch was recorded on the neighboring button.

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 [21]. We obtained similar results in our study and observed this behavior to be more significant when navigating than when selecting a candidate. With a button height of 18.4mm, the average position for a candidate selection was 10mm down from the top of the button, while the average position for a navigation was down 12.6 mm. We believe there are two possible explanations for touches to navigation buttons occurring closer to the bottom of the button. First, navigation buttons are near the bottom of the screen, so voters will be moving their fingers towards the bottom of the screen before pressing a navigation button. Also, the candidate buttons have no separation between their neighbors, while navigation buttons are completely surrounded by background space. This may result in voters using less caution when touching a navigation button than when selecting candidates.

Our results indicate that miscalibration strongly effects navigation in the case of a downward miscalibration. In nine tests with downward offsets ranging from 2.6 to 9.2mm, voters touched the background 193 times, out of 924 total touches for a background touch rate of 20.89%. In 159(82.4%) of those cases, the voter hit a navigation button after one or more background touches. On the other hand, in the five miscalibration tests with upward offsets, 33 out of 338 touches were background touches, only three(9.1%) of which were followed by navigation. In the control case 78 touches out of 1823 were to the background with 47(60.25%) followed by navigation. In the control case 4.53% of touches were to the background. Figure 7 shows these results along with the overall miss rates for all three cases.

We have not yet collected sufficient data to conclusively establish a connection between miscalibration and an increase in selection changes. 15 subjects have experienced

Figure 7. Miscalibration Background Touch Rates



a miscalibration test, five of which were upward miscalibration, and 10 of which were downward miscalibration. Considering an average touch position for candidate selection of 10.8mm from the top, an upward offset of 2.6mm would leave the average touch well within the bounds of a button. We expect that larger offsets would create a higher rate of selection changes.

The increased miss rate demonstrated by our results suggest that voting system logs should record the rate at which voters miss valid targets. While our results with a small sample are inconclusive regarding changes in selection indicating miscalibration, Moffatt's results indicate that changes in candidate selection should also be logged.

It is useful to examine of the distribution of the Y coordinate of touches. 1823 touches from both candidate and navigation buttons were recorded in the control, dishonest, and compressed ballot tests to form the distribution shown in Figure 8. Insensitivity and calibration groups were excluded. The distribution demonstrates that 97% all touches occur in the bottom two thirds the button, while the peak of the distribution is up 6.1mm from the bottom of the button. Given this distribution, it is not surprising that downward miscalibration creates higher error rates than upward miscalibration.

The tight distribution of touch positions allows us to use the average location of a touch within a button 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

Figure 8. Touch Location Distribution

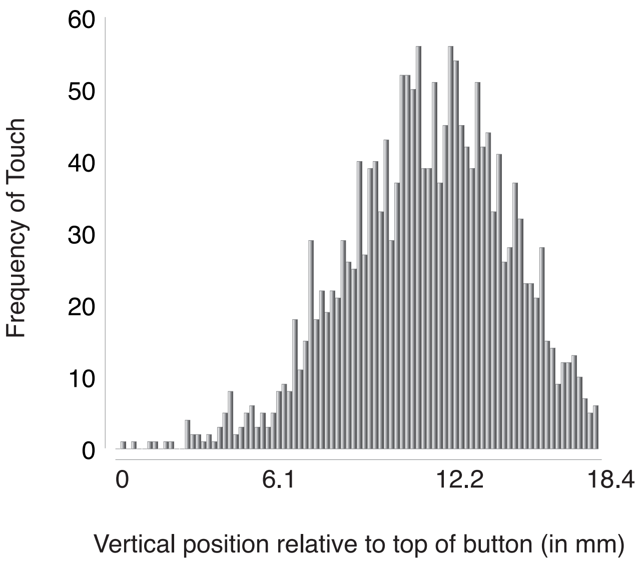
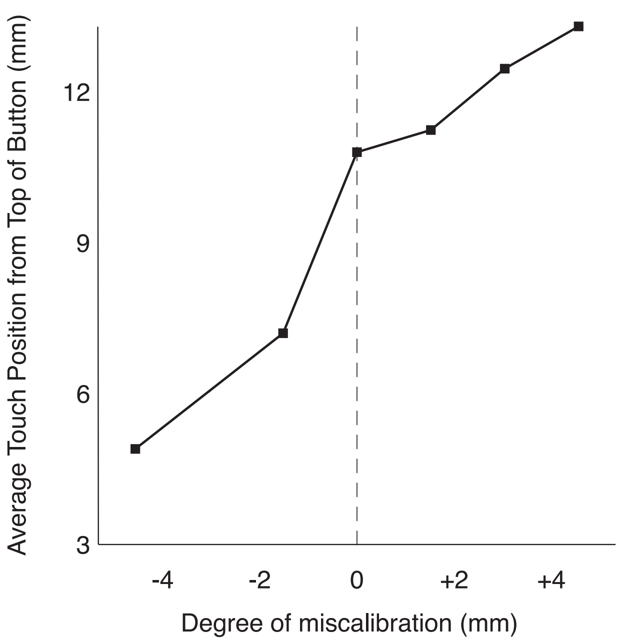


Figure 9. Average Calibration Touch Position



be recorded closer to the bottom of the button, as shown in Figure IX. We therefore recommend that relative coordinates be logged to help identify touchscreen miscalibration.

X. Future Work

This work presently contains only a preliminary analysis of data since we are still actively collecting data. To increase confidence in the findings of our study we indend to expand the total number of subjects by approximately another 100. As we increase the sample size for the study we hope to more accurately match the study demographics to those of the average voting population. Once we have a larger sample for the study, we indend to expand the analysis of our hypotheses to control for demographics factors such as age and computer experience.

Improvements also need to be made to maintain voter privacy with our logging scheme. Since there are a minimum number of navigation events needed to cast a ballot, it is possible to infer from which races a voter abstains if the voter only performs the minimum number of forward navigation button presses. It would be possible to eliminate this vulnerability by adding an 'abstain' button to each race. Our current analysis indicates that it is not necessary to log linear navigation events, which means it would be sufficient to only record navigation from the summary screen to contest update pages.

We intend to further examine the apparent increase in selections on compressed ballots. Previous research by David Kimball et. al. suggested that displaying too many choices on a screen can increase undervotes. However, he also found that undervotes increased on longer ballots, suggesting that voters may become fatigued. [16]. These results indicate that while placing one race per page may reduce undervotes due to banner blindness, it may increase undervotes to to voter fatigue. We suggest that more research be done to examine the role of ballot design. This further research should examine the role placing multiple similar races on a page has on undervote rates.

Finally we need to develop specific decision rules to diagnose the types of problems discussed in this paper.

XI. Conclusion

Our study demonstrates that several different types of voter behavior correspond to malfunctions in voting systems. Increased navigation to update pages corresponds with dishonesty in the summary screen. Placing multiple races on a page appears to either increase or decrease undervotes depending on the circumstance. Artificial touchscreen insensitivity leads to an increase in the length of time a voter touches the screen after system feedback and decreases the rate at which voters review races. Screen miscalibration increases the background touch rate and changes the average touch in the direction of the miscalibration.

These findings lead us to recommend several new requirements for voting system event logs that increase the likelyhood a post-election audit could properly identify abnormalities in voting system behavior. Recording the frequency at which voters navigate back to certain races from the review screen would help identify races which were undervoted due to poor ballot design. It could also indicate the presence of a dishonest ballot design. The interval of time between visual feedback from a touch, and the finger release should be recorded as an indicator of insensitivity. Background touch rates, candidate deselection and relative touch coordinates should be recorded to help identify touch screen miscalibration.

REFERENCES

[1] 107th U. S. Congress, "Help America Vote Act of 2002," October 2002. [Online]. Available: http://www.fec.gov/hava/law_ext.txt

[2] Committee on National Security Systems, "National information assurance glossary," Tech. Rep. 4009, June 2006.

[3] Federal Election Commission, "Performance and test standards for punchard, marksense and direct recording electronic voting systems," Tech. Rep., 1990.

[4] Federal Election Commission, "Voting systems performance and test standards," Tech. Rep., 2002.

[5] U.S. Election Assistance Commission, "Voluntary voting system guidelines," Tech. Rep., 2005.

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

[7] 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.

[8] A. Yasinsac, D. Wagner, M. Bishop, T. Baker, B. de Medeiros, G. Tyson, M. Shamos, and M. Burmester, "Software review and security analysis of the ES&S iVotronic 8.0.1.2 voting machine firmware, final report," Security and Assurance in Information Technology Laboratory, Florida State University, Tech. Rep., February 2007. [Online]. Available: http://election.dos.state.fl.us/pdf/FinalAudRepSAIT.pdf

[9] J. McTammany, "Balloting device," U.S. Patent 502 743, 1893.

[10] A. Cordero and D. Wagner, "Replayable voting machine audit logs," in *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop*, 2008.

[11] K. P. Yee. Pvote. [Online]. Available: http://pvote.org/

[12] N. L. McClure, R. D. Wieland, V. L. Babbitt, and R. A. Nichols, "Precinct voting system," U.S. Patent 705 012 000, 2006.

[13] Johnson County, Iowa Auditor's Office, "November 4, 2008 presidential election," November 2008. [Online]. Available: http://www.johnson-county.com/auditor/returns/0811elec.htm

[14] M. Pagendarm and H. Schaumburg, "Why are users banner-blind? The impact of navigation style on the perception of web banners," *Journal of Digital Information*, vol. 2, no. 1, 2001.

[15] J. P. Benway, "Banner blindness: What searching users notice and do not notice on the world wide web," Ph.D. dissertation, Rice University, April 1999.

[16] D. C. Kimball and M. Kropf, "Voting technology, ballot measures, and residual votes," *American Politics Research*, vol. 36, no. 4, pp. 479–509, 2008.

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

[18] S. P. Everett, "The usability of electronic voting machines and how votes can be changed without detection," Ph.D. dissertation, Rice University, 2007.

[19] I. S. MacKenzie and C. Ware, "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*, 1993.

[20] D. W. Jones, "Observations and recommendations on pre-election testing in Miami-Dade County," September 2004. [Online]. Available: http://www.cs.uiowa.edu/ jones/voting/miamitest.pdf

[21] 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.