

# Document Analysis Support for the Manual Auditing of Elections

Daniel Lopresti    Xiang Zhou    Xiaolei Huang    Gang Tan  
Department of Computer Science and Engineering  
Lehigh University  
Bethlehem, PA 18015, USA

## Abstract

*Recent developments have resulted in dramatic changes in the way elections are conducted, both in the United States and around the world. Well-publicized flaws in the security of electronic voting systems have led to a push for the use of verifiable paper records in the election process. In this paper, we describe the application of document analysis techniques to facilitate the manual auditing of elections, both to assure the reliability of the final outcome as well as to help reconcile the differences that may arise between repeated scans of the same ballot. We show how techniques developed for document duplicate detection can be applied to this problem, and present experimental results that demonstrate the efficacy of our approach. Related issues concerning machine support for the auditing of elections are also discussed.*

## 1. Introduction

Recent events in our history demonstrate that the transition to electronic voting can be a rocky one. Because of the unusual demands in running a nationwide election that may, in fact, be administered at the local level across tens of thousands of precincts, there are numerous opportunities for problems to arise. Inaccurate vote tallies caused by software bugs, malicious attacks, and other sorts of failures are a serious concern for those placed in charge and, indeed, for all citizens.

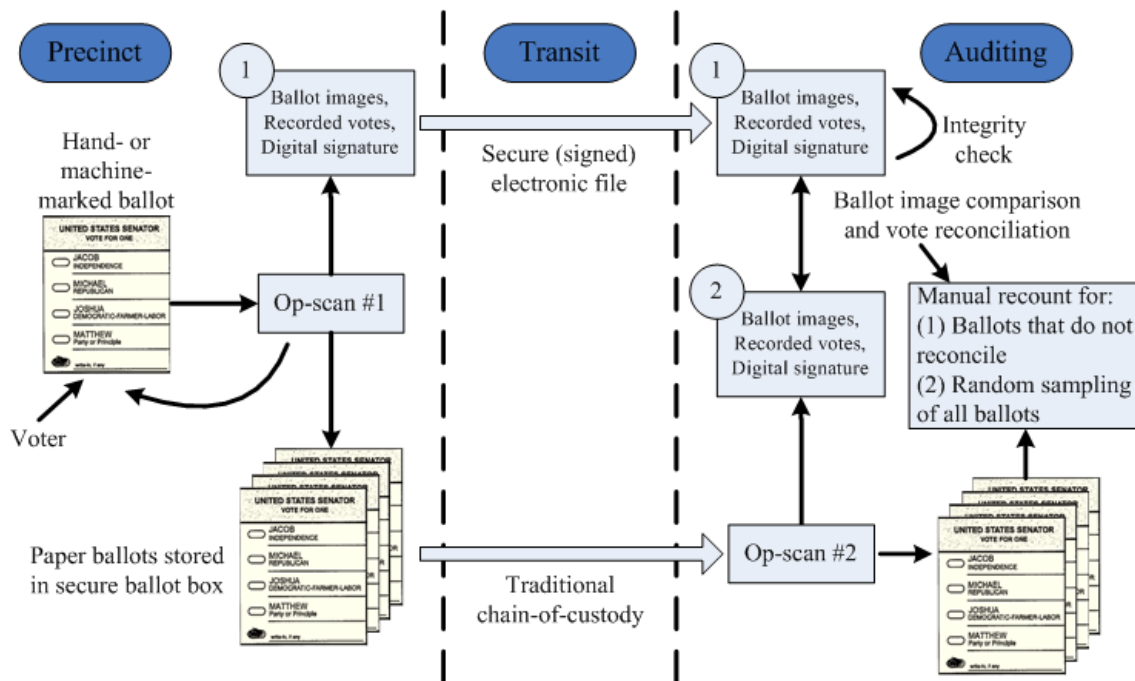
Although accurate tallies are crucial to a trustworthy electoral process, they are almost impossible to ensure with 100% certainty. As in other applications, redundancy is a potential solution, although not necessarily the best and only solution. These concerns have led computer security experts and voting advocates to argue for the use of the Voter-Verified Paper Ballot (VVPB), which provides valuable forensic evidence for use when problems or disputes occur. Paper is accepted to provide a degree of assurance. For example, the Help America Vote Act (HAVA)

requires that all Direct-Recording Electronic (DRE) voting machines produce a paper audit trail. According to a survey conducted among 523 voters in our home state of Pennsylvania [2], over 81% of the respondents stated that they believe such verification is important.

While the use of paper records brings fundamental benefits to the election process, auditing (recounting) all of the ballots in a given geographic area can be expensive, both in time and money. As noted in [4], for a trial recount of a DRE paper trail performed in Cobb County, Georgia, workers took an average of 5 minutes per ballot to audit 976 votes at a total cost of nearly \$3,000. Regardless of the underlying protocol, it is clear that hand recounts are neither rapid nor especially accurate.

In a recent paper, Calandrino, *et al.* propose an approach for conducting much more efficiently the random manual audits mandated by law in many states [3]. This clever scheme employs a second scan of the paper ballots after they have been shuffled to preserve voter anonymity. At the same time this scan is made, the ballots are marked with a unique serial number so that they can be associated with their interpretations, *i.e.*, the machine recognition results for the markings on the ballot. A random sampling is then performed so that a subset of the ballots can be manually recounted to confirm that the original tally and, by extension, the declared winner of the election are correct with high probability.

This model is much more efficient than performing a full precinct-level recount, but Calandrino, *et al.* do not address one lingering issue in their work: what happens if there is a discrepancy between the first scan of the ballot, which takes place at the precinct (and is, in fact, the only scan that is under the purview of the voter), and the second scan, which takes place at the time of the audit? As those who work in document image analysis know, it is quite common for multiple scans of the same document to produce different results (see, *e.g.*, [9]). For two tallies of the same election to differ could cause concern and raise doubts about the true winner and the trustworthiness of the process. In such a case, all of the ballots may need to be checked by hand.



**Figure 1. Proposed scheme for reconciling tallies for precinct and recount scans.**

In this paper, we build on the previous work by Calandrino, *et al.*, supplementing their approach so that it is possible to reconcile all of the differences between the two sets of ballot scans. We propose to use an existing technique for detecting duplicates in document image databases, and illustrate how this might work in practice. These ideas could be incorporated in ballot-based election audits with little additional expense to further increase confidence in the election’s outcome.

The rest of the paper is organized as follows. Section 2 presents the framework of our system. In Section 3, we describe our approach to feature extraction and ballot comparison via modified Hausdorff distance. The results of preliminary experiments are given in Section 4. Section 5 concludes with a discussion of future work.

## 2. System Framework

As already suggested, our scheme makes possible exactly the same ballot-based manual recount as Calandrino, *et al.* In addition, it permits us to reconcile any differences that may exist between the original precinct tally and the second tally by manually recounting ballots that were interpreted differently. An overview is depicted in Figure 1.

Briefly, the approach works as follow:

1. At the precinct level, the paper ballots are filled out by voters and fed into a scanner for the first time. Here,

rather than record only the votes as proposed in [3], we also record the images of the ballots.

2. Then the paper ballots are physically transported to the audit site through a traditional chain-of-custody mechanism, while the electronic file is transmitted over a secure channel using a digital signature for protection.
3. The paper ballots are scanned and read a second time to conduct the recount. They are also given unique ID’s at this point.
4. A manual recount of a ballot is triggered when: (a) the two scans of the ballot do not reconcile; or (b) the ballot is chosen for recount as part of the statistical random sampling process.

After the second (recount) scan, we progress through all of the ballots, one-by-one, considering the set of purported duplicates from the original (election) scan for each ballot. Each ballot image in the recount must be matched to at least one ballot image from the original election (the threshold for matching is relaxed until at least one ballot is in the match set). Multiple potential matches are possible, however, if two ballots are marked similarly.

Let  $S$  be the set of ballot images from the first scan that match a given ballot image  $B$  from the second scan. A case-by-case analysis follows:

**Case I** The Scan 2 interpretation for  $B$  matches the Scan 1 interpretation. No recount required, although we may not realize this. There are two subcases:

**Subcase Ia** The interpretations match. All of the images in Set  $S$  have the same interpretation and it matches the Scan 2 interpretation for  $B$ . In this case, the decision is to not recount  $B$  and this is the correct decision (a “true miss”).

**Subcase Ib** There is a mismatch among the interpretations. At least one ballot in Set  $S$  has a different interpretation from the Scan 2 interpretation for  $B$ . In this case, the decision is to manually recount  $B$  and this is an incorrect decision – it leads to extra work, but does not hurt the results of the tally (a “false hit”).

**Case II** The Scan 2 interpretation for  $B$  does not match the Scan 1 interpretation. In this case, a manual recount is required, although we may not realize this. As before, there are two subcases:

**Subcase IIa** The interpretations match. All of the images in Set  $S$  have the same interpretation and it matches the Scan 2 interpretation for  $B$ , but not the Scan 1 interpretation for  $B$  (which we do not realize because the image for  $B$  is not in the set). In this case, the decision is to not recount  $B$  and this is an incorrect decision that prevents us from reconciling the two tallies (a “false miss”).

**Subcase IIb** There is a mismatch among the interpretations. At least one ballot in Set  $S$  has a different interpretation from the Scan 2 interpretation for  $B$ . The decision is to manually recount  $B$  and this is the correct decision (a “true hit”).

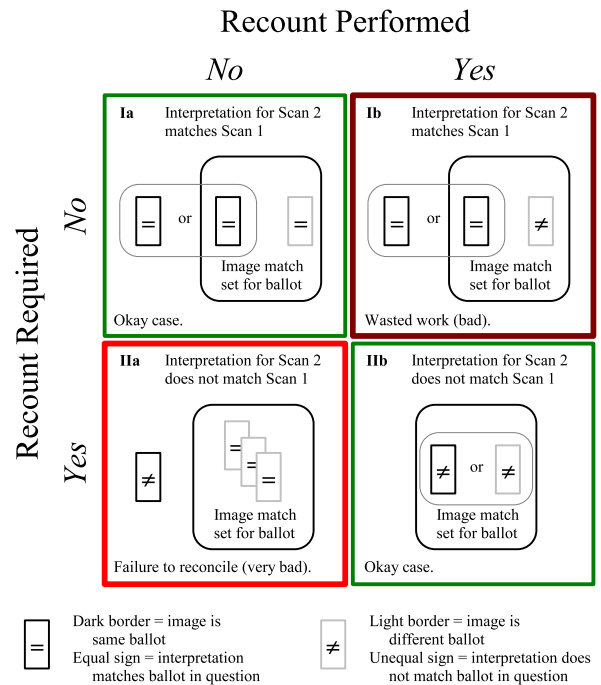
As indicated, the case that leads to extra work is Ib. The case that leads to failure in reconciling the tally is IIa. Our ultimate goal is to avoid the latter while minimizing occurrences of the former. These cases are depicted in Figure 2.

### 3. Duplicate Document Image Detection

To identify which scans may correspond to the same physical ballot, we turn to techniques developed for the duplicate document detection problem in the image domain.

#### 3.1. Pre-processing

After initial pre-processing of the ballot images, we use the Iterative Closest Point algorithm (ICP) proposed by Besl and McKay [1] to register the images from both scans to a template ballot. By doing so now, we only need to perform this step once, which saves computation time.



**Figure 2. Recount case analysis.**

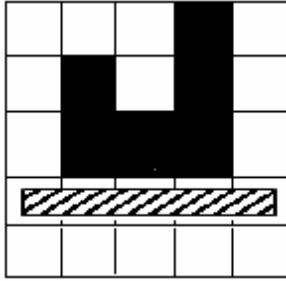
The procedures can be described as follows:

1. Extract feature points from a predefined area on the two ballot images using a Harris Corner Detector [5].
2. Sort the Hessian Matrix values of all detected corners to obtain the largest  $n$  feature points.
3. Form two feature vectors using these points, one for each of the images.
4. Pass the feature vectors on to the ICP algorithm.

Testing shows that the proposed pre-processing method is quite accurate. We scanned 10 paper ballots (out of 100) with intentionally large skew angles and translations. The registration algorithm handled all of them with high accuracy. For example, the algorithm might output 6.48 degrees when the actual skew angle is 6.5 degrees.

#### 3.2. Extracting pass codes

After pre-processing of the ballot images, we extract “pass codes” using Hull’s algorithm [6]. Pass codes are employed in CCITT compression to encode black or white runs of pixels on a given row which are not connected to a run of the same color on an adjacent row. We utilize this property to extract distinct features for the ballot images.



**Figure 3. Passcode example.**

For each row in the ballot image, we scan left-to-right to see if there exists a longer black or white run of opposite color to the one just above it. If there is one, we mark the middle of this run as one of the pass codes. In Figure 3, there is a 3-pixel black run in Row 3, and a 5-pixel white run just below it (these pixels are marked by a shaded rectangle). Hence, the pixel in the middle of this white run (the third pixel in Row 4) will be one of the pass codes extracted.

This feature works well for describing ballot images because it provides an accurate representation of the details on the page, *e.g.*, it can capture white holes in filled oval targets and also dark markings (noise) around or within the targets.

### 3.3. Modified Hausdorff distance

We chose to use modified Hausdorff distance as the metric for evaluating image similarity. We adapted this somewhat to our particular application. The steps for generating modified Hausdorff distance are as follows:

1. Let two ballot images overlap one another. For each pass code in Ballot A, find the nearest pass code in Ballot B within a  $40 \times 40$  pixel square. The distance between these two pass codes falls in the interval  $[0, 29)$ .
2. For each pass code in Ballot A, if the distance satisfies  $d \in [k, k + 1)$ , increment the corresponding variable  $\text{binA}[k]$ ;
3. If there is no such corresponding pass code, increment  $\text{binA}[29]$ ;
4. Repeat Steps 1 to 3 for Ballot B. Ultimately, we get  $\text{binA}[0]$  to  $\text{binA}[29]$ , and  $\text{binB}[0]$  to  $\text{binB}[29]$ . These values form a 60-dimensional feature vector.

## 4. Experimental Evaluation

In this preliminary study, we evaluate our approach on two specific tasks. The first is to determine whether it can detect the ballots that need to be recounted (*i.e.*, ballots where

the Scan 2 interpretation differs from the Scan 1 interpretation). Then we consider the problem of identifying missing and/or added ballots, a possible sign of wholesale election fraud.

### 4.1. Reconciling precinct and audit scans

Our template is the State General Election Ballot from Minnesota in 2006. We printed copies of the ballot which were then randomly marked by more than 50 students from various departments at Lehigh. A total of 2,130 bitonal images were created from these ballots at an overall size of 2,552 pixels by 3,300 pixels. TIF images were scanned at 300 dpi and encoded using the CCITT group 4 standard, each totalling about 100KB. Every Scan 2 ballot had a corresponding match scanned from the same paper ballot within the Scan 1 set.

Our implementation is based on Ubuntu Linux 8.04 supported by an Intel Core2 Duo running at 1.8 GHz with 2 gigabytes of RAM. After considering several features and similarity metrics, we eventually settled on the following measure. We count the number of pass codes in Image 2 which are within a distance  $d$  of each pass code in Image 1. Then we divide this value by the total number of pass codes in Image 2 to get a ratio. The same procedure is repeated from the other direction. We treat the geometric average of these two values as the similarity between ballot images.

By our earlier discussion, we require a manual recount of a ballot when Case Ib or Iib arises. We now examine how many ballots we need to check by hand.

We first selected 100 ballots from our dataset to simulate a precinct. For each image in Scan 2, the program needs to compare it with all of the images in Scan 1 to determine Set  $S$ . So there are a total of 10,000 image comparisons that need to be performed. Among the 100 ballot images, 98 have the same interpretation in the two scans, while two ballots were interpreted differently. The goal, then, is to find these two ballots while at the same time recounting as few of the images as possible.

For each image in the second scan, we chose the most similar image from the original scan to form the Set  $S$ . Since the only variable in the above algorithm is  $d$ , we varied this from 3 pixels to 10 pixels to find the best value. In doing so, we determined that the algorithm performs best when  $d$  equals 7. Under this setting, 84 out of 100 images fall into Case Ia, 14 fall into Case Ib, and the final two are in Case Iib. This means that we only need to recount 16 ballots (out of the 100 total) to capture all of the discrepancies in our mock election.

From our experiments, we have found that Hausdorff distance is a good metric to use. The rationales include:

- It is relatively insensitive to small perturbations.

- Intuitively, if the Hausdorff distance is  $d$ , then every point in Shape A must be within a distance  $d$  of some point in Shape B, and vice versa.
- Portions of one shape can be compared to another.

Simplicity and modest computational cost are two more advantages.

## 4.2. Missing and Added Ballots

Another possible situation arises when ballots are missing or added between the first and second scans. In such cases, we note that the similarity should be high if the image in Scan 1 has a corresponding match in Scan 2. Conversely, if there is a missing ballot in Scan 2, its corresponding image in Scan 1 should not have strong similarity to any of the images in Scan 2. To test this, we scanned 100 paper ballots, then randomly deleted four ballots and scanned the remaining 96 ballots a second time.

Acting on the above assumption, for each image in Scan 2, the program extracts the most similar image from Scan 1. If any ballot image in Scan 1 is not represented among the extracted images, this may be a missing ballot in the later (recount) scan. We can then check manually to determine whether this is really a missing ballot.

In our tests, the program returned 19 suspicious images from the 100 in the original set. Fortunately, all four of the missing ballots were in this set, although roughly one-fifth of the total ballots have to be checked. Using the same procedure, we can also identify added ballots by simply exchanging the roles of the original and recount scan sets.

## 5. Conclusions

In this paper, we have built on the work of Calandrino, *et al.* by recording the images of scanned ballots and using them to help reconcile any discrepancies between the precinct and recount tallies. We described a reliable framework for the problem and presented some preliminary experimental results. Based on our studies, it appears that modified Hausdorff distance is a good metric to use in this case. The net result will be more trustworthy voting when using paper ballots.

Future work will be focused on finding better solutions for dealing with added and missing ballots and conducting experiments on degraded ballot images. We close by noting that there are a rich variety of document analysis problems arising in the context of electronic voting research. The PERFECT project has as its goal the development of more accurate mark recognition algorithms for op-scan systems [7, 8].

## 6. Acknowledgments

This work was supported in part by the National Science Foundation under award number NSF-0716368. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

## References

- [1] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [2] C. Borick, D. Lopresti, and Z. Munson. 2006 Survey of Public Attitudes Toward Electronic Voting in Pennsylvania. Technical Report LU-CSE-06-35, Department of Computer Science and Engineering, October 3 2006. Lehigh University / Muhlenberg College Institute of Public Opinion.
- [3] J. A. Calandrino, J. A. Halderman, and E. W. Felten. Machine-assisted election auditing. In *Proceedings of the USENIX / Accurate Workshop on Electronic Voting Technology*, Boston, MA, 2007.
- [4] S. Dunn. Voter verifiable paper audit trail pilot project, November 2006.  
[http://www.gaforverifiedvoting.org/docs/Cobb\\_county\\_pilot\\_report.pdf](http://www.gaforverifiedvoting.org/docs/Cobb_county_pilot_report.pdf).
- [5] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.
- [6] J. J. Hull. Document image similarity and equivalence detection. *International Journal on Document Analysis and Recognition*, 1(1):37–42, 1998.
- [7] D. Lopresti, G. Nagy, and E. B. Smith. A document analysis system for supporting electronic voting research. In *Proceedings of the Eighth IAPR Workshop on Document Analysis Systems*, pages 167–174, Nara, Japan, September 2008.
- [8] Paper and Electronic Records for Elections: Cultivating Trust (PERFECT), 2009.  
<http://perfect.cse.lehigh.edu/>.
- [9] J. Zhou and D. Lopresti. Repeated sampling to improve classifier accuracy. In *Proceedings of the IAPR Workshop on Machine Vision Applications*, pages 346–351, Kawasaki, Japan, December 1994.