Breaking Assumptions: Distinguishing Between Seemingly Identical Items Using Cheap Sensors

William Banks Clarkson

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Advisor: Edward W. Felten

June 2012

Abstract

Intuitively, we understand that physical items are distinct — we are able to pick up and manipulate them. Yet, we often treat similar items as if they are indistinguishable from one another. Undue reliance on the assumption that *seemingly* identical items can be treated as if they are identical can lead to unintended, negative consequences. Conversely, the ability to distinguish between seemingly identical objects often enables new applications. This thesis examines three artifacts that carry with them false assumptions about their indistinguishability: blank sheets of paper, ‘anonymous’ bubble forms, and loudspeakers.

Blank sheets of paper, e.g. those in a ream of copy paper, are often treated as if they are identical to one another. We develop a new method to identify sheets of paper by measuring their inherently unique surface texture using commodity equipment. This permits a number of applications, e.g. counterfeit currency detection, and has important implications for paper-based elections, which are discussed in detail.

Next, we turn to optical-scan bubble-forms which often rely on the assumption that they do not reveal the respondent’s identity. We demonstrate that individuals tend to mark bubbles in distinctive ways, unintentionally conveying their identity. This has important implications for anonymous surveys and the publication of completed ballots after an election. We describe a number of mitigation techniques and procedural changes to limit the risk of accidentally revealing identifying information.

The final artifact, loudspeakers, are increasingly ubiquitous devices designed to accurately reproduce an audio signal. It is commonly assumed that multiple instances of ‘identical’ loudspeakers will generate the same output given identical inputs. This assumption is false. We demonstrate that individual loudspeakers, even those of the same make and model, induce unique distortions on the generated sound, identifying the individual loudspeaker. We develop methods to identify loudspeakers, enabling a new method of device authentication.

By examining the common underlying assumptions of each artifact, we develop a common methodology used in identifying distinguishing features. This general framework is successfully applied to each artifact, suggesting that other seemingly identical objects may become distinguishable in the future.

Acknowledgements

I would first like to thank parents, Mom, Dad and Dan. Your incredible advice, to always pursue what interests you most, has been invaluable over the course of my education. To my family, thank you for your unwavering support and encouragement, both to attempt a graduate education and in every other aspect of my life. Without you, this dissertation would never have been possible.

To my wife, Vicki. You have been with me from the very beginning of this adventure. Your unwavering encouragement, even in moments of self-doubt, have made this dissertation possible. I can not express to you how much your love and support means to me. I am excited for the next chapter of our adventure together.

While there are too many to mention, I would like to thank my some of my fellow Computer Science Graduate Students: Joe Calandrino, Ian Davey, Ari Feldman, Shirley Gaw, Michael Golightly, J. Alex Halderman, Nick Jones, Josh Kroll, Tim Lee, Wyatt Lloyd, Jeff Terrace, Harlan Yu and Bill Zeller. Each of you has inspired me in numerous ways, and each of you made my time at Princeton a true joy, both personally and professionally.

The Center for Information Technology Policy (CITP) has been a wonderful resource, full of intelligent, enlightening, and thought-provoking individuals, panels and conferences. The experiences and thoughtful discussions were incredibly interesting and helped to shape my outlook on the potential impact of technology on governance and society at large. I look forward to the immense impact that CITP will all have in the years to come.

Ed Felten, I cannot thank you enough for the many hours of guidance and for the encouragement to attempt projects that, at their conception, seemed unlikely to succeed. I would also like to thank Adam Finkelstein, who was a source of passion and inspiration to attempt projects outside of my normal comfort zone.

This work was supported in part by Google Inc.

To my parents (Mom, Dad and Dan) and to my amazing wife, Vicki.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We all make simplifying assumptions. Such assumptions are often necessary to function effectively in an increasingly complex world. However, making a simplifying assumption can lead to an important, unintended, negative side effect: the loss of information. If we fail to periodically re-examine these assumptions, e.g. when environmental conditions change, we risk undue reliance on abstractions and applications that are based on ultimately false assumptions. Within the past decade, the landscape of what is measurable has fundamentally changed. The dramatic increase in accuracy and plummeting cost of sensors demands that some common assumptions be re-examined.

The goal of this thesis is to challenge one common assumption — that *seemingly* identical items are identical. In many instances, this particular assumption minimizes daily mental friction. For example, if you printed out this thesis, you likely didn't give a moment's thought to the pages upon which it is printed. You probably treated each page in the ream of paper as if it is indistinguishable from all the others. This particular assumption of indistinguishability is common. Yet, we intuitively understand that each blank page is distinct, and therefore unique. In most circumstances, this assumption does not have important implications. In others, relying on this ultimately false assumption can lead to undesirable outcomes.

Most of us do not stop and ask: What is it makes each sheet of paper unique? How hard is it to tell them apart? In what other situations might the ability to distinguish between pages be relevant? More generally, do other items carry with them similar assumptions of indistinguishability? If so, what properties must an object have to make it unique? How can we measurethese properties? If we can identify each individual object, what are the implications in the 'real' world? Questions like these are important to ask, especially with the rapidly increasing availability of cheap sensors that allow highly accurate measurement of physical properties. This thesis is an attempt to provide answers to some of these questions — and not just about blank sheets of paper.

1.1 Historical Context

The identification of items based on inherent physical attributes has a long and storied history. Ballistic fingerprinting is one of the first examples of a item being uniquely identified based on inherent physical attributes. Beginning in 1835, when a suspect confessed to a crime after markings on a bullet retrieved from the victim matched those from his gun, the science of identifying the firearm responsible for firing a bullet has advanced rapidly [48]. In 1925, Calvin Goddard brought the science of ballistics fingerprinting into the mainstream through a Popular Science article describing a method of re-identifying bullets using his newly invented comparison microscope [42]. We still use this approach today, as illustrated in Figure 1.1 which depicts markings on two bullets, pictured side-by-side, which were fired through the same gun. That individual bullets exhibit markings linking them to a particular firearm is common knowledge today. But it was not always this way. The widespread availability of the microscope changed the landscape of what was measurable.

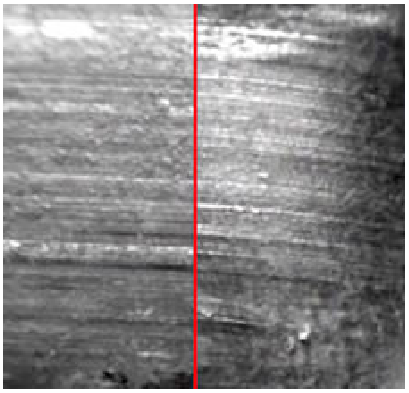


Figure 1.1: Comparison between two bullets fired from the same gun [75]. This comparison technique was popularized by Calvin Goddard using his comparison microscope.

Typewriters are another example of a class of objects that at one time were presumed to be identical to one another. Typewriters were one of the first widely available mechanical devices for which a fingerprint could be derived [47]. Throughout the early 1900s, there were fewer than a dozen domestic manufacturers of typewriters. This allowed seasoned examiners to determine the make and model based only on visually inspecting characteristic variations in a sample document. As the quality and number of manufacturers grew, the difficulty in visually determining the source typewriter paved the way for more analytical identification methods. These methods included measuring variations in alignment, ink density due to defects in the ribbon, or broken type faces. The ability to 'fingerprint' the source of a particular document allowed for new applications, particularly the ability to link a document to the responsible typewriter in court cases. The most public example involved Alger Hiss, who was convicted of perjury based, in part, on evidence tying him to a particular typewriter [91].

Examples like ballistic fingerprinting and typewriter identification are numerous. Table 1.1 contains some items with use-cases that implicitly rely on assumptions about

- Bullets
- Typewriters
- Desktop Scanners & Cameras (Same Make and Model)
- RFID Tags
- Home Appliances (Same Make and Model)
- Toilets
- Blank Sheets of Paper
- ‘Anonymous’ Bubble-Forms
- Loudspeakers

Table 1.1: List of items once thought to be indistinguishable from one another.

their indistinguishability. Items within each category are often used interchangeably with other *seemingly* identical items. For example, it was believed that images taken with digital cameras or scanned via desktop scanners did not reveal the specific device that generated the image. It turns out that variations in the light sensors allow the resulting images to be traced back to their source devices [56]. RFID Tags are often used in situations where they are meant to convey unique identifiers, e.g. to identify a passport or credit card. While RFID communications can be encrypted, preventing the secret identifier from being revealed to non-key holders, the physical inconsistencies between ‘identical’ RFID tags allow each physical tag to be identified based on easily measurable variations [107]. This results in a rather effective method of tracking people, e.g. shoppers in a shopping mall, based on the RFID tags in their wallet, clothing or medical device. Conversely, it prevents a tag from being truly copyable — as the features that make each RFID tag identifiable, also prevent it from being copied perfectly. Home appliances, such as televisions, computers, refrigerators and light bulbs all induce characteristic noise onto a home's power lines. This allows each device to be identified by its particular noise pattern using a cheap sensor attached to any power outlet. Interestingly, a similar property holds true for toilets. [37, 45].

The availability of the microscope and other measurement techniques allowed for the development of ballistic fingerprinting and increased the sophistication with which typewriters could be identified. As evidenced by the examples in Table 1.1, I argue that the increasing availability of highly accurate, cheap sensors provides new opportunities to re-evaluate widely held assumptions about whether a particular item is indistinguishable from seemingly identical items. Even for items which were previously identifiable, the widespread availability of cheap sensors significantly reduces the barrier, allowing everyday sensors to identify these items. This fundamental shift facilitates new applications. This thesis provides a number of examples where the assumption that *seemingly* identical objects are identical turns out to be false and discusses the implications (both positive and negative) and describes a number of new applications.

1.2 Organization

This thesis provides several new examples of items that were once thought to be indistinguishable, but are actually identifiable with high accuracy. We begin in Chapter 2 with the introduction of a General Framework that is helpful in examining seemingly identical objects. In the following chapters, the General Framework provides a rough outline for examining different items whose physical attributes are measurable using a cheap sensor.

Chapter 3 shows how blank sheets of copy paper can be re-identified with 100% accuracy, even when treated harshly. Each sheet of paper has an inherently unique surface texture. We develop methods to measure this surface texture using only a commodity desktop scanner, allowing each sheet of paper to be uniquely identified. This presents a number of applications and implications, e.g. providing a cheap and resilient method to identify counterfeit currency and documents, or allowing any document to be traced through time and space without modification. For elections, this result has a significant, negative implication; the ability to re-identify an individual's ballot. It also provides new opportunities to identify fraud — enabling a method to detect ballot box stuffing. The implications of paper fingerprinting for elections are discussed in Chapter 4.

Even without measuring the surface texture of a sheet of paper, it may still be possible to identify a respondent's ballot. Bubble-forms, which are commonly used on anonymous surveys, paper ballots, and standardized tests, often come with the assumption that they cannot reveal the respondent's identity, beyond what is conveyed by the particular answer selections. Chapter 5 re-evaluates this assumption and shows that individuals tend to mark bubbles in distinctive ways, unintentionally conveying their identity through the simple act of marking a bubble. This surprising result can have important implications, undermining the privacy guarantees of certain applications, while simultaneously allowing for new applications. The news is not all bad, Section 5.4 provides suggestions on how to mitigate the privacy risk.

Loudspeakers are an increasingly ubiquitous device and are currently built into a number of everyday devices, e.g. cell phones, televisions, cars, and home theatre systems. Many of us treat loudspeakers as simple devices that generate music, ringtones, or soundtracks, implicitly assuming that loudspeakers of the same make and model will generate the same output when provided with the same input signal (e.g. all produce the same output when playing the same Lady Gaga song). In truth, each loudspeaker distorts the input signal in a characteristic and measurable way, even if imperceptible to the ear. This distortion can be captured by a computer's microphone and some signal processing. Chapter 6 studies the ability to measure this distortion to identify a particular loudspeaker. Interestingly, this characteristic can have positive implications for device authentication and could eventually lead to increased fidelity of audio generated by low-quality loudspeakers.

In the Chapter 7, I discuss where I believe this area of research is heading and the broad implications of these, and similar techniques, for society. Everything is uniquely identifiable — and the ever increasing availability of sensors will result in an increasing number of uniquely identifiable items which have important policy implications going forward.

This thesis contains some previously published results. In particular, many of the methods described in Chapter 3 were published in 2009 at the IEEE Symposium on Security and Privacy [26]. Chapter 4, which discusses the implications of paper fingerprinting for elections, contains certain results which were published in the Proceedings of EVT/WOTE in 2009 [22]. Finally, the majority of Chapter 5 was published at USENIX Security in 2011 [23].

Chapter 2

General Framework and Background

Distinguishing between objects based on inherent physical attributes is an increasingly common research area. Previous research repeatedly follows an implicit framework when distinguishing between seemingly identical artifacts. Yet, it has not been formally codified. This commonly used framework is increasingly relevant due to a combination of factors including the increasing accuracy, plummeting cost, and widespread availability of a variety of sensors, along with recent advancements in machine learning and the exponentially decreasing cost of storage and computational power. These factors make the identification of seemingly identical items more viable now than ever before, and increasingly so in the future. This chapter introduces the General Framework which is helpful when evaluating seemingly identical items and serves as the rough outline for the structures of Chapters 3, 5 and 6.

2.1 General Framework

Each chapter begins with an item that is seemingly identical to other items. In each case, we identify a particular physical attribute and measure it using an inexpensive

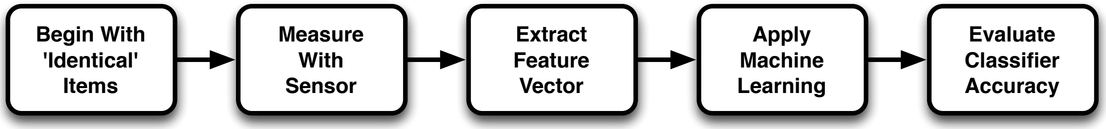


Figure 2.1: General Framework

sensor. Next, we extract a feature vector from the measured attribute which captures the unique properties of each object. We use machine learning techniques to train a classifier that is able to identify patterns between feature vectors and distinguish items from other seemingly identical ones. In the final stage, we evaluate the method of distinguishing between seemingly identical items. In hindsight, the General Framework informed the process of selecting, measuring, and evaluating each of the artifacts discussed in the subsequent chapters. See Figure 2.1.

2.1.1 Begin with 'Identical' Items

When similar objects are treated as if they are identical, there is likely a use-case that implicitly relies on this property, i.e. $similar \implies identical$. In each chapter, we begin with an artifact that carries with it some assumptions about its indistinguishability. We challenge these assumptions and measure a physical property, ideally without modifying the item in any way. The major challenge is finding a feature that is easily and consistently measurable using a cheap sensor.

2.1.2 Measure with Sensor

At the molecular level, every item is composed of a unique arrangement of molecules. These arrangements have implications at larger scales of measurement which are easier to measure with cheap sensors. The trade off is at what scale are we able to reliably and cheaply measure an effect of these distinct molecular arrangements. A good starting point is the macroscopic properties that distinguish similar kinds of physical objects. These properties are often the same ones used determine the quality of an item. For example, what differentiates card stock and copy paper — the thickness and texture. What differentiates an expensive loudspeaker and the speaker in your phone — the range and intensity of frequencies that can be accurately reproduced. It is often the case that differences in these physical features are likely be present, but attenuated, in seemingly identical objects, e.g. multiple sheets of copy paper or multiple instances of a particular model of loudspeaker. Once a feature is identified, it becomes an iterative process to find a sensor and method to generate feature vectors that extracts the physical attribute and identifies an item consistently.

2.1.3 Extract Feature Vector

Once a physical attribute can be measured, it is often helpful to distill the raw measurements down to a more concise representation. The distilled version of these attributes is called a feature vector. Feature vectors are designed to robustly capture the most unique and distinguishing characteristics of an object. Taking multiple measurements of the same object and generating feature vectors which are labeled with the object they came from can be advantageous at abstracting the most salient components of the feature vector. Once the feature vectors are generated and labeled, it is common to use machine learning algorithms to determine which features are most distinguishing for a particular item.

2.1.4 Apply Machine Learning

Machine learning is a branch of Artificial Intelligence (AI). In practice, machine learning focuses not on artificial thought, but on different ways to classify objects based on patterns extracted from a set of input examples. In traditional machine learning research, a set of examples, called a training set, is used by a machine learning algorithm to infer indicative patterns. Once indicative patterns are learned, a second set of examples, called the test set, is used to measure the ability with which the machine learning algorithm captured the most distinguishing features. See Section 2.2 for high-level background on the machine learning algorithms used in this thesis.

2.1.5 Evaluate Classifier Accuracy

After a classifier is trained, we evaluate the ability of the classifier to distinguish between items in a variety of circumstances. The goal is to be able to distinguish items with perfect accuracy, at least under ideal conditions, and reasonably well when items are treated with less care. Once an item is distinguishable from seemingly identical items, the most interesting part begins; evaluating the implications. The ability to identify a particular physical item may also lead to applications previously thought of as unachievable.

2.2 Common Background

This thesis utilizes a variety of methods from Computer Graphics, Machine Learning, Signal Analysis and Physics. Combining techniques from such diverse fields means that many readers will not be familiar with the all of the different methods used. I describe at a high level, some common concepts that will be useful in subsequent chapters. This overview is meant only to give the intuition into how and why these techniques are useful, not to serve as a definitive reference.

Machine learning algorithms come in two main flavors, supervised and unsupervised. Supervised learning algorithms use a set of examples, each of which are labeled with a particular class. The machine learning algorithm determines which features of each example, and related variations and correlations of features, are most indicative of a particular class. The intuition is that the set of examples in the training set will allow the algorithm to derive a rule that can make successful predictions when provided with similar inputs in the future. These future inputs are likely to be slightly different, and the ability of the algorithm to recognize complex patterns is an important factor in determining the ultimate success of a machine learning algorithm.

In contrast to supervised learning algorithms, unsupervised learning algorithms are given a set of examples which are not labeled with any particular class. Depending on the application, unsupervised machine learning algorithms may be explicitly told the number of classes to identify, or may algorithmically determine how to segregate the examples into sets of similar items. The algorithm systematically sifts through the examples, placing each example into a set with other similar examples. The number and characteristics of each set are often determined by maximal separation between the feature vectors of each example. When complete, each set contains examples which are similar to one another, and maximally dissimilar from all other examples. The intuition is that once classified, future examples which are most similar to items in a particular set likely came from the same underlying class. Unsupervised learning algorithms are often used in situations where it is impractical to generate a set of examples for which the true labels are known.

In the next three chapters, both kinds of learning algorithms are utilized. We make heavy use of a machine learning workbench, Weka, which provides a large number of pre-built implementations of common supervised and unsupervised machine learning algorithms [46]. We use Support Vector Machines and in particular Sequential Minimal Optimization (SMO), which is an efficient method of training SVMs. We also make use of Principal Component Analysis which is a way to reduce the size of feature vectors. SVM and PCA are examples of supervised and unsupervised learning algorithms, respectively, which we provide an overview of in the next section.

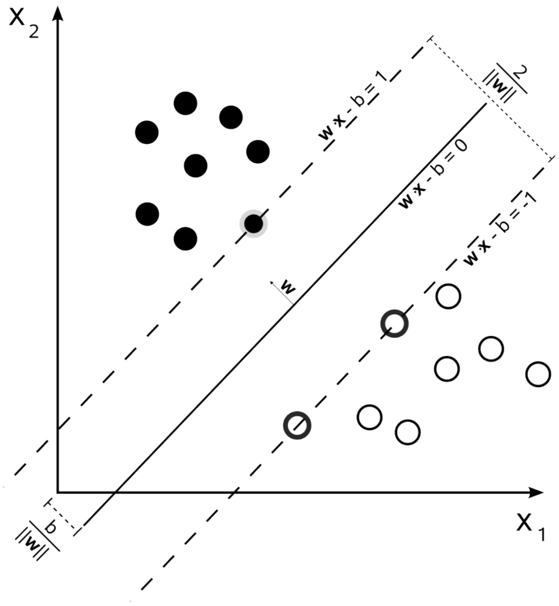


Figure 2.2: Support Vector Machines construct a hyperplane in a two-dimensional space. The hyperplane $w * x - b = 0$ maximizes the separation between classes.

2.2.1 Support Vector Machines

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm commonly used for classification tasks. SVMs take a set of examples with associated labels as input (the training set). Each example consists of an input vector x and a label y. Traditionally the set of training data consists of pairs (x_i, y_i) where x_i is a vector in a p-dimensional subspace, e.g. it consists of p numbers, and y_i is either 1 or -1. SVMs are binary classifiers, only able to distinguish between two classes. Multi-class classifiers are built by combining a number of binary classifiers.

At a high level, SVMs divide the input feature vector space by creating one or more hyperplanes. These (p-1)-dimensional hyperplanes separate training examples based on which side of the hyperplane each example falls on. The goal is to calculate a set of hyperplanes such that all points of a particular class fall on one side of the hyperplane. The exact parameters of each hyperplane are chosen such that there is maximum distance between each of the examples in the training set and the hyperplane surface. Figure 2.2 provides a conceptual visualization of a 2-dimensional example.

SVMs can be trained using a number of different training algorithms. The task of training an SVM can always be accomplished by solving a large linear quadratic programming optimization problem — in practice this can be challenging to solve. Sequential Minimal Optimization (SMO) allows SVMs to be trained in a much more efficient manner [82].

2.2.2 Weka

Weka, the Waikato Environment for Knowledge Analysis, is a software workbench designed to aid in data mining and machine learning experiments [46]. Weka contains a wide variety of pre-built machine learning and data mining algorithms including implementations of SVM and SMO. We made heavy use of these, and other features of Weka in our analysis.

2.2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a widely used technique in a number of fields including computer graphics and machine learning. From a practical perspective, PCA takes high-dimensionality data and distills it down to a lower dimensionality — often while maintaining much of the original information. PCA works by taking a set of (possibly) correlated inputs, and generating a set of statistically independent outputs. These outputs are called the principal components. The intuition is that each high-dimensional input can be approximated as a linear combination of a smaller number of principal components.

Calculating the principal components from a set of inputs is straightforward. We begin with a set of K input examples, which are selected so as to be representative of the type and variety of inputs the algorithm is likely to see. Each of the K examples is treated as a column in an M x K matrix. We assume each example is M-dimensional. To calculate the principal components, each of the M dimensions is normalized appropriately, and the mean is subtracted. Next, the covariance matrix and the set of eigenvectors V are calculated that diagonalize the covariance matrix [52].

The eigenvectors of this matrix are the principal components, and the eigenvalues are the weights of each eigenvector. The weights are interpreted as the contribution of each eigenvector to the overall information content. By taking the top-N eigenvectors, as determined by the sorted order of their corresponding eigenvalues, the information content of the source can be distilled down to an arbitrarily long N-dimensional vector. In practice, the dimensionality reduction from M to N is substantial. One common rule-of-thumb is to choose N such that the 75% of the information is captured in the top-N eigenvectors. See Figures 2.3 and 2.4 for visual representations of what eigenvectors and approximations of a set of input images look like.

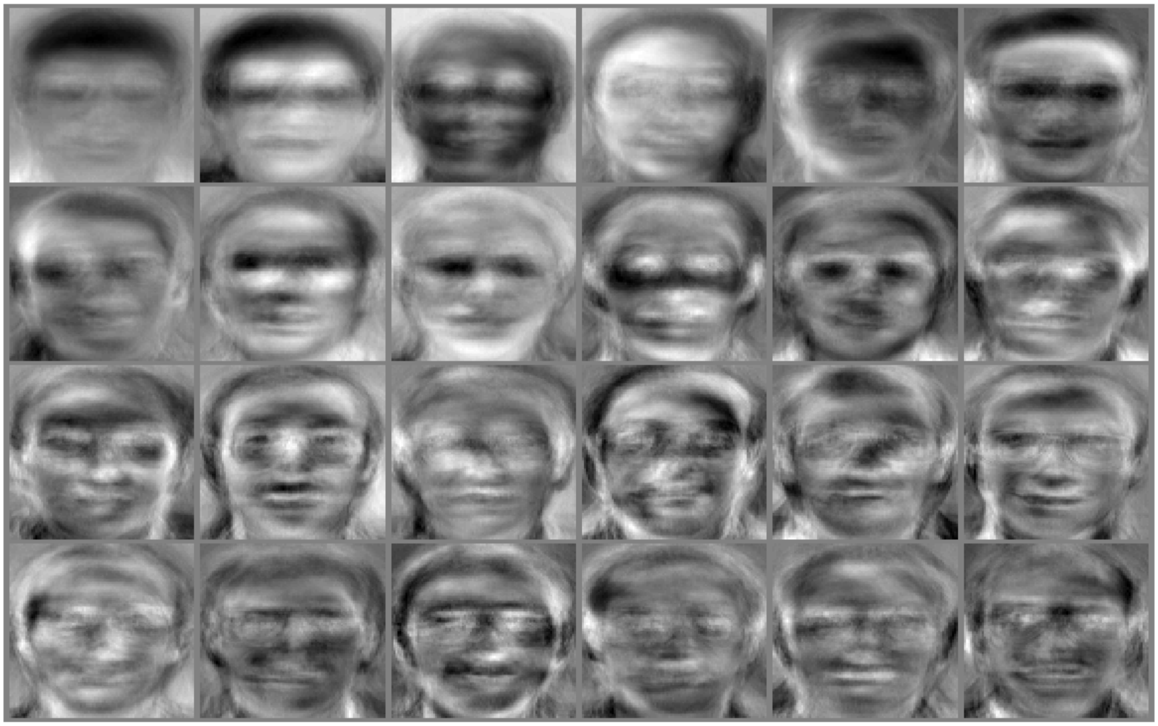


Figure 2.3: The top 25 principal components taken from a representative set of face images. For clarity, we treat each principal component, which is just an eigenvector, as an image, which we call an eigenface. An image of a face can be approximated by a linear combination of these 25 eigenfaces. See Figure 2.4 for examples. Source: Chris DeCoro, cs.princeton.edu/˜cdecoro/

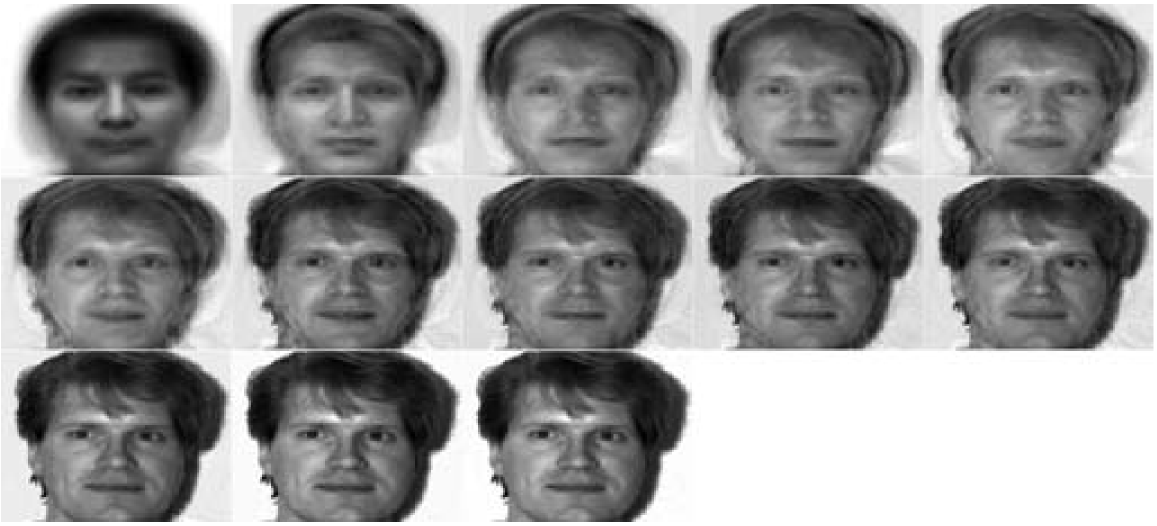


Figure 2.4: Moving from left to right, 8 additional principal components, are used to approximate the input image. The face begins to resemble the original after approximately 64 principal components. This number is significantly less than the number of original pixels in the input image. Source: Chris DeCoro, cs.princeton.edu/˜cdecoro/

Figure 2.3 depicts the top 25 eigenvectors (treated as images) from a representative set of face images. By taking linear combinations of these eigenvectors, an input image that contains a face can be approximated using a 25-dimensional feature vector. This 25-dimensional vector is significantly smaller than the original image, and contains a significant percentage of the information in the original image. See Figure 2.4 for examples of approximations of the input image with an increasing number of eigenvectors.

For a formal treatment of Principal Component Analysis, see [52].

Chapter 3

Fingerprinting Blank Paper Using Commodity Scanners

Blank sheets of paper are often treated as if they are indistinguishable from one another. For example, each sheet of paper in a ream of copy paper is typically treated no differently than any other. This is often a result of the mistaken belief that a blank sheet of paper is a flat, empty canvas upon which content is placed, and that once content is added, it is the content that makes a document unique. In fact, when viewed up close, the surface of a sheet of paper is not perfectly flat, but is a tangled mat of wood fibers with a rich three-dimensional texture that is highly random and difficult to reproduce. See Figure 3.1. This rich texture provides an excellent feature that can distinguish one sheet of paper from another.

In this chapter, we show that the surface texture of paper can be estimated using only a flatbed scanner coupled with appropriate software, and that this feature is robust against rough treatment—such as printing or scribbling on the document or soaking it in water—and adversarial counterfeiting. Under normal conditions, this technique can identify documents with near-perfect accuracy and a negligible false positive rate.

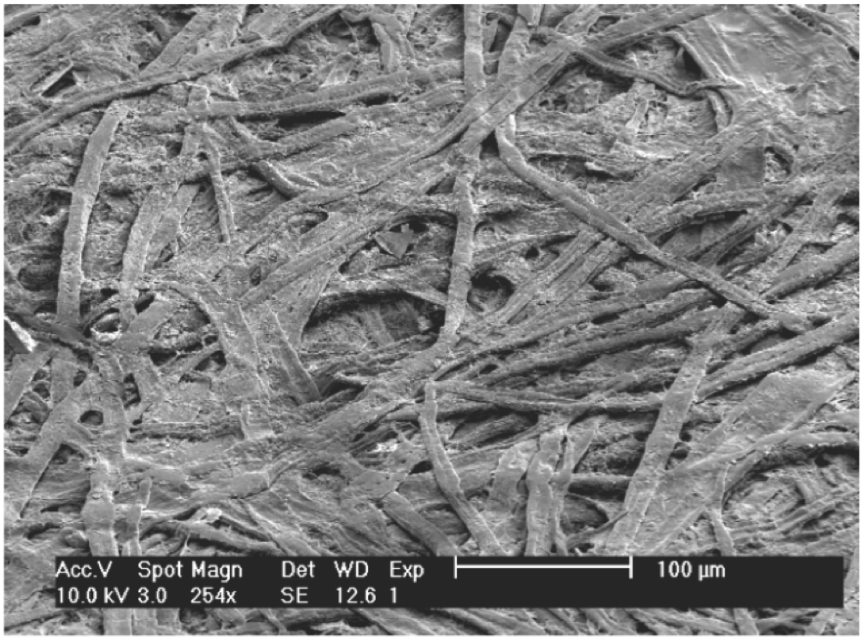


Figure 3.1: The surface texture of paper viewed under a microscope at 254x magnification. Each sheet of paper is made up of a complex pattern of paper fibers that are created during the manufacturing process.

It has long been known how to authenticate the *content* printed on a page by using cryptographic methods such as digital signatures. We address a different problem: how to authenticate the paper itself. For some kinds of documents, such as currency and tickets, it matters not only that the content is unaltered but also that the document is a genuine original rather than a copy or forgery. Physical document authentication has many applications, which we discuss in Sections 3.6 and 3.7. Some of these applications may be harmful; for example, our method allows re-identification of supposedly anonymous surveys and paper ballots.

In contrast with previous efforts, our technique measures paper's 3-D texture, requires no exotic equipment, produces a concise document fingerprint, does not require modifying the document, and may be applied to blank paper before content is printed. Previous systems lack one or more of these properties. For example, Laser Surface Authentication [17] requires a costly laser microscope to image paper texture, while the technique proposed by Zhu et al. [109] focuses on ink splatter caused by

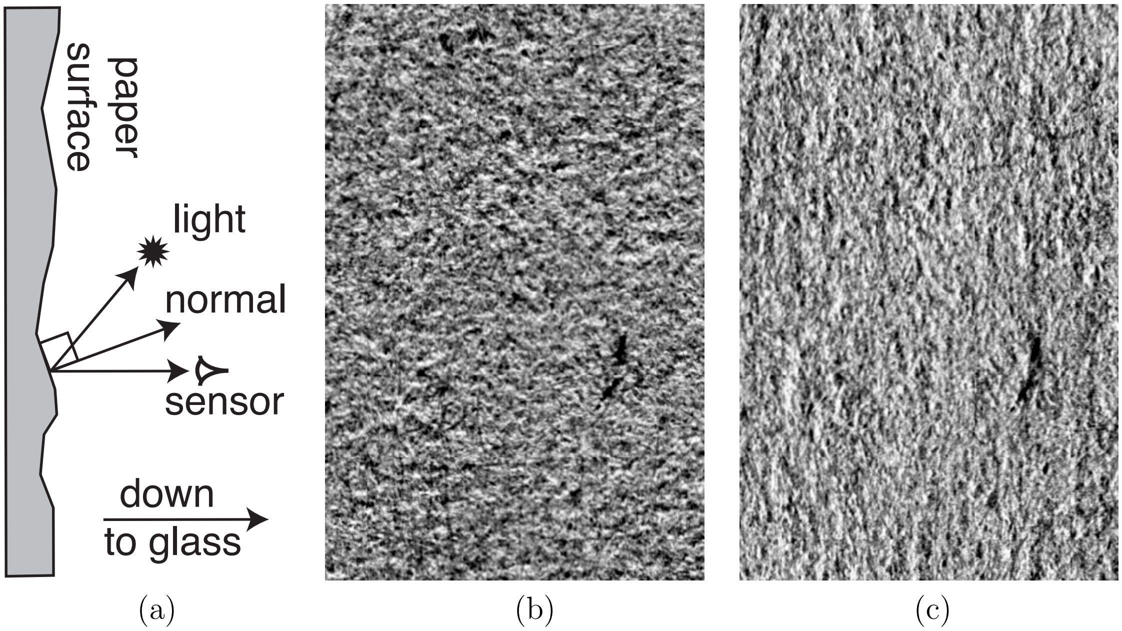
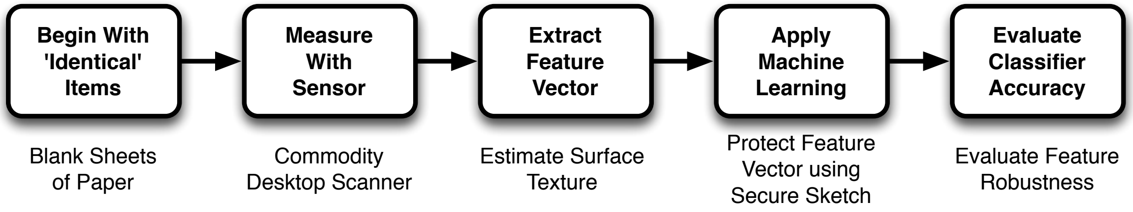


Figure 3.2: Since the surface of a sheet of paper is not perfectly flat, a scanner will produce a different image depending on the orientation of the page. The light reaching the sensor depends on the relative angles of the light source and surface normal, *(a)*. A 10 mm tall region of a document scanned from top to bottom, *(b)*, appears different from the same region scanned from left to right, *(c)*. By combining (b) and (c) we can estimate the 3-D texture.

randomness in the printing process and requires the paper to be printed with known content prior to fingerprinting. We discuss these and other related work in Section 3.1.

3.0.4 Organization

The physical document authentication technique described in this chapter roughly follows the outline of the General Framework, see Figure 3.3. We begin in Section 3.2 (Stage 2 of the General Framework), by describing how we can use a scanner to estimate a document's surface texture. Scanners normally measure only the color of a document, but by scanning the paper several times, at specific orientations, we can estimate the shape of the surface (see Figure 3.2). In Section 3.2.2 (Stage 3 of the General Framework), we condense the surface texture into a concise feature vector, which robustly identifies the page. In this instance, instead of applying machine learning techniques, we use a technique called a secure sketch to generate a fingerprint that reveals little meaningful information about the feature vector. The resulting fingerprint can be printed on the document (e.g., as a 2-D bar code) or stored in a database. The verification procedure is similar to the fingerprinting process, with a different final stage that verifies that the generated feature vector is correct.



Begin With 'Identical' Items
Measure With Sensor
Extract Feature Vector
Apply Machine Learning
Evaluate Classifier Accuracy
Blank Sheets of Paper
Commodity Desktop Scanner
Estimate Surface Texture
Protect Feature Vector using Secure Sketch
Evaluate Feature Robustness

Figure 3.3: Application of the General Framework introduced in Chapter 2 to identifying blank sheets of paper based on their unique surface texture.

We designed our technique to satisfy several security and usability goals:

- **Uniqueness** Every document should be identifiable and distinguishable from all others.
- **Consistency** A fingerprint should be verifiable by multiple parties over the lifetime of the document.
- **Conciseness** Document fingerprints should be short and easily computable.
- **Robustness** It should be possible to verify a fingerprinted document even if it has been subjected to harsh treatment.
- **Resistance to Forgery** It should be very difficult or costly for an adversary to forge a document by coercing a second document to express the same fingerprint as an original.

Sections 3.3–3.5 evaluate our system in terms of these same goals (Stage 5 of the General Framework).

3.1 Related Work

The Fiberfingerprint system of Metois et al. first introduced the notion of using surface texture to uniquely identify a document [68]. Employing a custom device, Fiberfingerprint measures "inhomogeneities in the substrate" of a document, from which a unique identifier is derived. The system employs alignment marks that are added to the document in order to orient the verification system, and requires a specialized hardware device rather than a commodity scanner.

Laser Surface Authentication is a technique that measures the texture of a page using a high-powered laser microscope [17]. Creating and verifying fingerprints in their system requires access to this specialized device, which may be prohibitively expensive for many users. Their system also requires that the verifier be online, which may rule out applications such as third-party ticket verification.

A recent patent application by Cowburn and Buchanan describes using a commodity scanner to identify documents [29]. This method does not measure the normal vector field of a document, but rather uses scans from multiple orientations in order to extract other additional information. The feature vector used by Cowburn and Buchanan is not concise, and their fingerprint is not secure. An adversary with access to the fingerprint is able to easily discover the surface texture of the document, possibly making forgery less difficult.

Zhu et al. focus on identifying "non-repeatable randomness existing in the printing process" [109]. They generate a fingerprint from the random ink splatter that occurs around the edges of any features printed on a page. Unlike our scheme, their method can only be applied after a document has been printed. Furthermore, their implementation requires modifying the original document by printing a known target pattern.

Our method is an improvement over previous work because we measure the surface texture of a document without the requirement of expensive equipment. We utilize

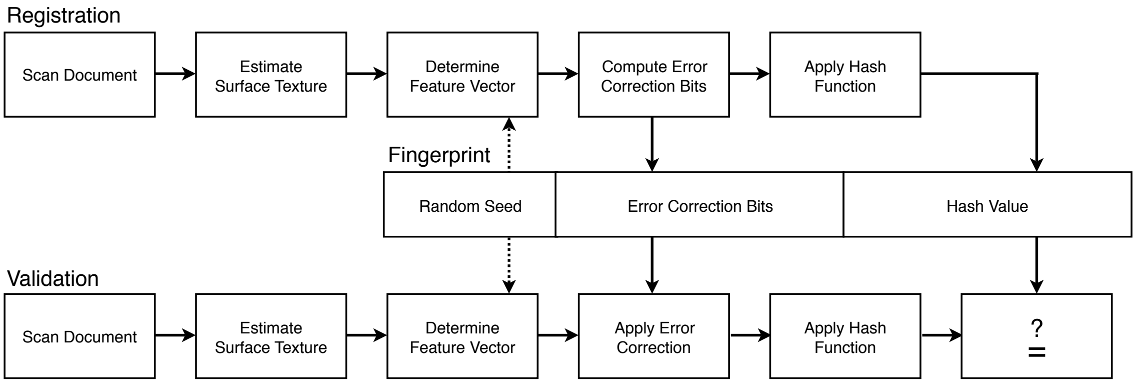


Figure 3.4: Registration and validation pipelines. *Registration:* Our method creates a fingerprint that consists of a hash value, error correction information, and a random seed. *Validation:* A document is authenticated if a newly computed hash value is identical to the one stored in the fingerprint. The stored error correction information is used to correct potentially faulty bits in the feature vector.

the unique fiber structure as identified and relied upon by Metois et al., Cowburn and Buchanan, and Zhu et al. but do so without modifying the document in any way. Our method allows documents to be fingerprinted before or after content is printed. In fact, fingerprinting and tracking using our system can begin during the paper manufacturing process. We have also developed methods for hiding the target feature vector through the use of a secure sketch. This means a potential counterfeiter cannot learn what features he needs to reproduce from the fingerprint alone but would need access to the original document to even attempt a forgery.

3.2 Fingerprinting Process

Our fingerprinting process allows for registration and validation of a sheet of paper without the participation of a central registration authority. Depending on the application, a document's fingerprint can be stored in a database for future verification, or it can be printed on the document along with a digital signature, creating a self-verifying original. The fingerprint can be used to ascertain whether two documents share the same feature vector without revealing the registered feature vector itself.

The registration and validation processes are quite similar, as shown in Figure 3.4. In the registration process, we scan a document, estimate its three-dimensional surface texture, and generate a feature vector $\mathbf{V}$ that represents the document's unique texture. We consider two documents to be the same if they have similar feature vectors. To protect the feature vector and inhibit forgeries that might seek to reproduce an exact feature vector, the fingerprint contains only a one-way hash $H(\mathbf{V})$ of the extracted feature vector. To achieve robustness against measurement errors in the feature vector, the registration process derives error-correction information from $\mathbf{V}$ and stores it in the fingerprint in the form of a secure sketch. The fingerprint also contains a random seed to initialize the pseudorandom number generator used to compute the feature vector, as described in Section 3.2.2.

The validation process has no access to the original feature vector. Validating a document requires determining a document's feature vector anew, using the seed stored in the fingerprint. Validation assumes a potentially flawed raw feature vector $\widetilde{\mathbf{V}}'$ and uses the secure sketch to obtain an error corrected $\widetilde{\mathbf{V}}$, as described in Section 3.2.3. The candidate document is considered valid if this feature vector maps to the same hash value stored in the fingerprint—that is, if $H(\widetilde{\mathbf{V}}) = H(\mathbf{V})$. The remainder of this section discusses the registration and validation pipeline in detail.

3.2.1 Estimating document surface texture

To measure the surface texture of paper, we take inspiration from a technique commonly used in computer graphics, called photometric stereo. Photometric stereo is a technique for estimating the surface texture of an object by measuring slight variations in the intensity of light reflected by an object when illuminated from multiple orientations. [105]. To give some intuition into why this technique works, imagine a series of pictures of a mountain range taken in the morning, midday, and late afternoon from the same location. By comparing these images, and measuring variations in the shadows cast by the mountains, we might be able to determine the height of the mountains in the picture, or at a minimum where the peak is and what the general slope of the mountain is. Photometric stereo works in an analogous manner, except photometric stereo doesn't rely on identifying variations in shadows, but measures variations in the observed intensity of light reflected by an object across multiple observations to extract an estimate of the object's surface texture.

Figure 3.5 depicts a figurine, illuminated by a light source at four different locations, whose color appears slightly different depending on the orientation of the light source relative to the ball. These slight variations contain information about the figurine's surface texture. The question is, how can we apply photometric stereo to measure the surface texture of a sheet of paper using only an ordinary scanner?

To capture the surface texture of a document, we scan each document at four orientations: 0°, 90°, 180°, and 270°. This allows recovery of the surface orientation for every sampled surface point. Our procedure assumes that the reflection of light from the surface of paper is perfectly diffuse, which is an assumption that largely holds for near-orthogonal observation. Diffuse materials reflect a portion of the incident light that is proportional to the cosine of the angle between the direction of incidence and the surface normal—that is, proportional to the dot product of these normalized directions. This property is commonly exploited in *photometric stereo* to reconstruct surface normals from multiple images under varying illumination by a point light source [105]. Similarly, we apply photometric stereo to the four captured scans. Flatbed scanners, however, contain a linear light source, rather than a point source, which disallows the application of traditional photometric stereo. Brown et al. recently demonstrated how normals can be derived from flatbed scans under multiple orientations [16]. Their method, however, relies on an extensive calibration procedure, which would make it impractical for authentication purposes. Instead, we will

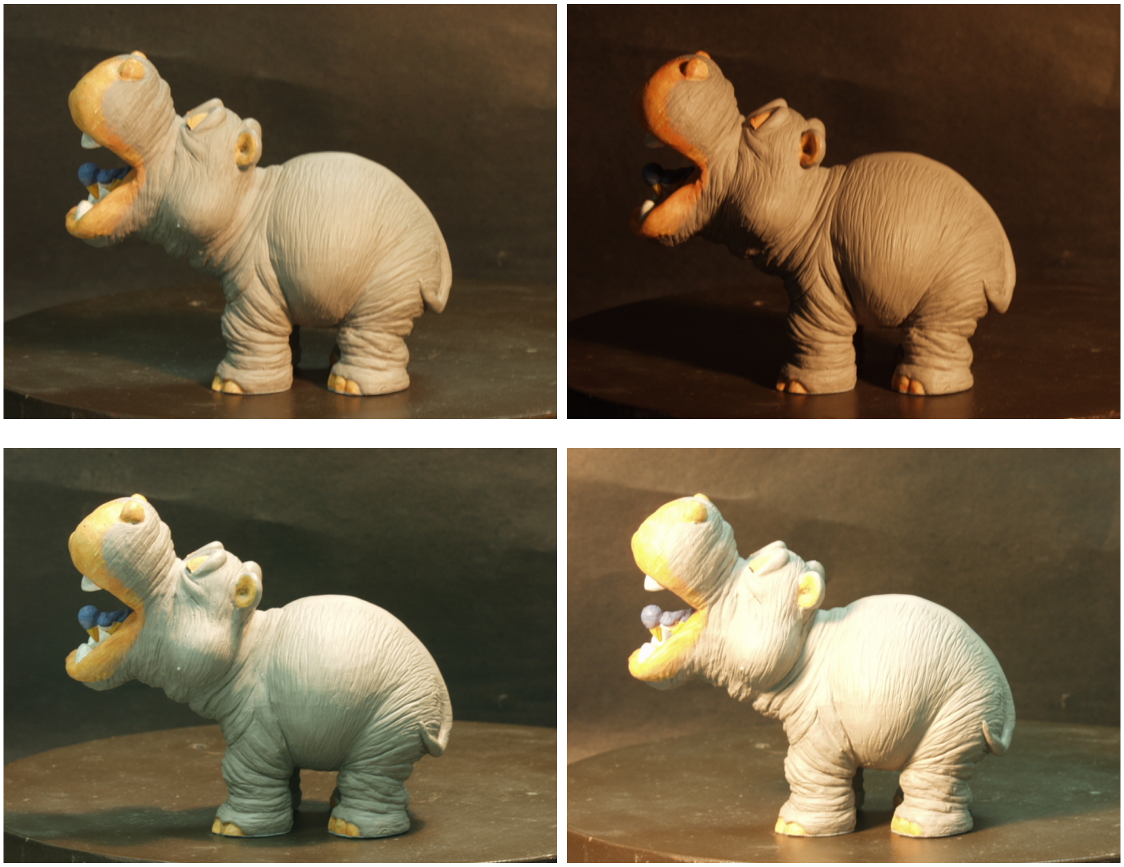


Figure 3.5: A hippo figurine is illuminated by the same light source from four different locations. Depending on the orientation of the light source relative to the figurine, the figurine reflects light with slightly different intensities. Photometric stereo exploits this phenomenon in order to estimate the surface texture of the figurine. Image source [12]

derive a novel photometric stereo solution for flatbed scanners, which provides us with information on surface orientation without the need for dedicated calibration.

Let us define a coordinate system for the paper and the scanner so that the paper lies in the xy-plane, the z-axis points away from the flatbed of the scanner, and the scanner's linear light source is parallel to the x-axis. See Figure 3.7. We approximate this light source by a line segment extending from x_1 to x_2. We further assume that the light source is offset with respect to the line on the paper imaged by the CCD sensor (see Figure 3.2(a)) by o_y in the y-direction and by o_z in the z-direction.

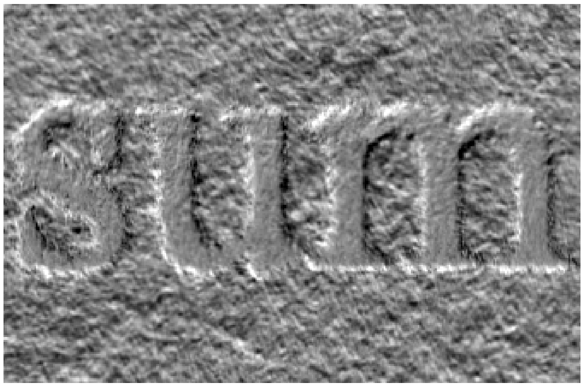


Figure 3.6: Difference image between two 1200 DPI scans showing the surface texture measured by the scanner in the y direction. Actual size: “sum”.

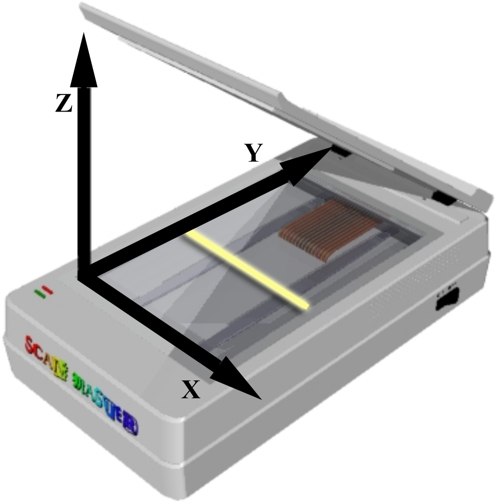


Figure 3.7: The coordinate system for the paper and scanner. The x-axis is parallel to the linear light source of the scanner, which moves along the y-axis. The z-axis points away from the surface of the scanner.

Each point on the paper has a normal $\mathbf{n}(x, y)$ and a diffuse color, or *albedo*, $\rho(x, y)$. Without loss of generality, we concentrate on a surface point at the origin of our coordinate system. The observed intensity of such a surface point is then:

$$I = \rho \int_{x_1}^{x_2} \left\langle \mathbf{n}, \frac{(x, o_y, o_z)^\top}{\|(x, o_y, o_z)^\top\|^3} \right\rangle \mathrm{d}x \quad = \rho \int_{x_1}^{x_2} \left\langle (n_x, n_y, n_z), \frac{(x, o_y, o_z)^\top}{\|(x, o_y, o_z)^\top\|^3} \right\rangle \mathrm{d}x \ , \quad (3.1)$$

which is the integral over all light diffusely reflected off that surface point and originating from points $(x, o_y, o_z)^\top$ along the linear light source. As every flatbed scanner is designed for even illumination, any limiting effects near ends of the light source are negligible and we shall ignore the integral limits in the remainder of this discussion.

Scanning the same surface point a second time with the paper rotated by 180° displaces the light source from o_y to $-o_y$. Subtracting the resulting two scans I_{0° and I_{180° from each other leads to:

$$\begin{aligned} d_y &= I_{0^\circ} - I_{180^\circ} \\ &= \rho \int \left\langle (n_x, n_y, n_z), \frac{(x, o_y, o_z)^\top}{\|(x, o_y, o_z)^\top\|^3} - \frac{(x, -o_y, o_z)^\top}{\|(x, -o_y, o_z)^\top\|^3} \right\rangle \mathrm{d}x \\ &= \rho \int \left\langle (n_x, n_y, n_z), \frac{(0, 2o_y, 0)^\top}{\|(x, o_y, o_z)^\top\|^3} \right\rangle \mathrm{d}x \\ &= n_y\, \rho \int \frac{2o_y}{\|(x, o_y, o_z)^\top\|^3} \mathrm{d}x \\ &= n_y\, \rho s\ . \end{aligned} \tag{3.2}$$

That is, the difference d_y yields an estimate of the y component n_y of the surface normal $\mathbf{n}$. The resulting value is multiplied by the albedo, ρ, and a fixed constant s that is dependent on the scanner geometry only. Analogously, $d_x = I_{270^\circ} - I_{90^\circ} = n_x\, \rho s$. With four scans we can determine, at each sample point, an effect of the surface normal's projection into to xy-plane, $\mathbf{n}_2 = (n_x, n_y)$, up to a scale. The factor s is assumed to be fairly constant across the page, and the remaining scale is given by the local surface reflectance ρ of the paper at any given location. The practical impact of ρ on the estimated surface normal is discussed in Section 3.5.

Application of equation (3.2) requires precise alignment of each surface point across all scans. To reduce the effect of alignment imprecision and to isolate frequencies of the document that are stable across scans and different scanners, we apply a low-pass filter to the document and down-sample it. In our experiments we scanned each document at 1200 SPI (samples per inch) and down-sampled it by a factor of eight, resulting in an effective resolution of 150 SPI. To align multiple scans, we calculated the homography between scans that minimized the least squares distance between them [54].[1]

After processing the four scans of a document, we recover the surface texture as a two-dimensional vector field with $\mathbf{d} = (d_x, d_y)^\top = \rho s\, \mathbf{n}_2$ defined at each location of the document.

3.2.2 Computing the feature vector

From this vector field $\mathbf{d}$ we determine the feature vector of the document. A good feature vector captures unique characteristics of a document, while at the same time being concise. We model the feature vector $\mathbf{V}$ as an N-bit vector of independent bits f_i whose values are derived from the surface normals of the document.

In contrast to previous approaches, we do not extract a feature vector from a single region of the document, but we compute the feature vector from a collection of representative subsections, *patches*, of the document. For documents down-sampled to 150 SPI, we choose square patches of 8×8 samples, centered at a series of random locations $\mathbf{p}_i$. For relatively even spacing we draw these locations from a Voronoi distribution [76]: we use the random seed stored in the fingerprint to initialize P pseudorandom start locations on the page and use Lloyd's Voronoi relaxation to obtain a set of locations distributed evenly across the document, as shown in Figure 3.8 [19].

In principle one could now directly compare the patches of a document A to corresponding patches in a document B in order to verify two documents as being the same. The disadvantages are that this requires access to the patches of B when verifying A, which would require an amount of storage prohibitive for offline applications,

[1] We printed a black box on each test page used in our experiments to identify the region to be fingerprinted. This box was used, out of convenience, to simplify the image alignment process. In practice, the black box is not necessary, and registration can occur by recording a few high-quality portions of the page, and aligning based on their locations [89]

and, more importantly, that it would reveal the original document's structure to a forger. Hence, we derive a compressed feature vector and store its hash along with a secure sketch to hide the feature vector from an adversary.

Each patch contains 64 2-D samples $\mathbf{d}_i$, $i = 1, \ldots, 64$, which we stack to create a patch vector $\mathbf{p} \in \mathbb{R}^{128}$. Each patch contributes T bits to the feature vector. We compute these feature bits f_i, $i = 1, \ldots, T$, by subsequently comparing the patch vector to T *template vectors* $\mathbf{t}_i$. The template vectors are a set of pseudorandomly chosen orthonormal vectors in $\mathbb{R}^{128}$ generated using the same seed that is used to determine patch locations: the $\mathbf{t}_i$ are initialized with vector components drawn from a $N(0, 1)$ distribution, followed by Gram-Schmidt orthonormalization. Each template vector can be interpreted as a template patch of 8×8 2-vectors denoting surface orientation.

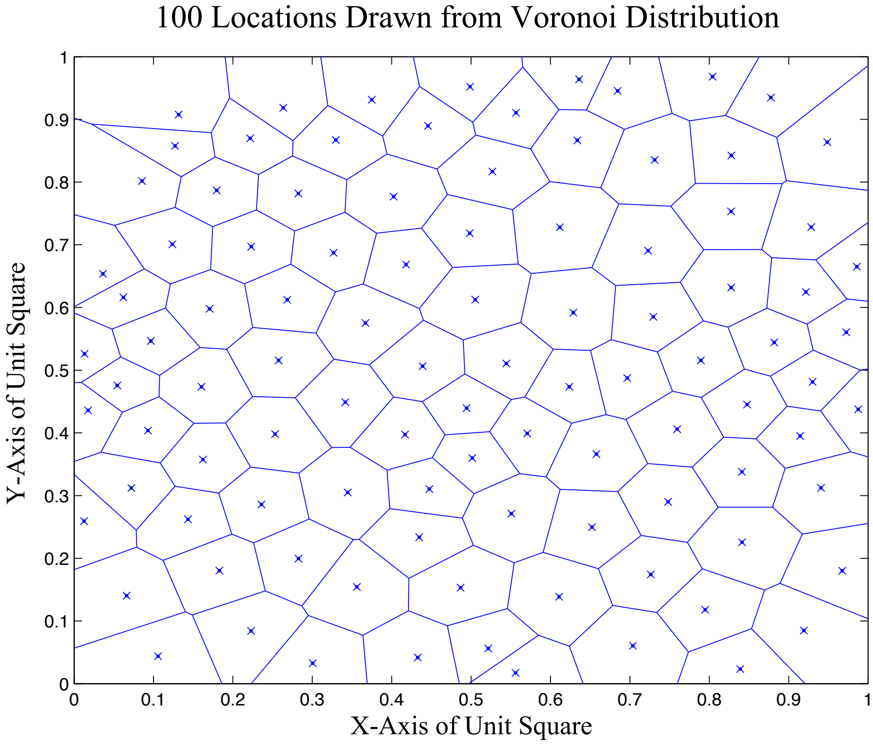


Figure 3.8: Sample Voronoi distribution of 100 points in the unit square. Voronoi distributions give relatively uniform coverage of a region, while simultaneously ensuring no overlap of patches.

The comparison is performed by correlating the patch vector $\mathbf{p}$ and each template vector $\mathbf{t}_i$; i.e., by computing the dot product $\langle \mathbf{p}, \mathbf{t}_i \rangle$. Positive correlation means that surface orientations in the patch and the template patch agree; negative correlation denotes mostly opposing surface orientations. The respective feature bit is determined by the sign of the correlation:

$$f_i = \frac{1 + \mathrm{sign}(\langle \mathbf{p}, \mathbf{t}_i \rangle)}{2} \ . \tag{3.3}$$

See Algorithm 1 for further illustration.

```
$\mathbf{V}$ = bool[$PT$]
Retrieve surface orientation vectors of document
Extract $P$ patches based on Voronoi distribution
for $p = 1$ to $P$ do
  template = generate new set of $T$ pseudo-
      random orthonormal template vectors
  for $i = 1$ to $T$ do
    $c = \langle$ patch[$p$], template[$i$] $\rangle$
    $f_{(p-1)P+i}$ = TRUE if $c > 0$
  end for
end for
```

Algorithm 1: Feature vector generation.

The number of independent bits that can be extracted from a patch in this way is limited and depends on the amount of information contained in a patch. A standard tool to characterize this amount of information is principal component analysis (PCA) [49]. We performed PCA on a large set of randomly chosen patches from different documents. The results show that for 8×8-patches at least 75% of the information can be expressed with only 32 principal components; that is, within a 32-dimensional subspace. We hence decided to restrict ourselves to $T = 32$ of 128 possible orthonormal template vectors . This restriction is intended to maintain independent bits in the feature vector, as additional template vectors are likely to result in increasingly correlated bits. The feature vector consists of 32, mostly independent, bits extracted from each patch. We use 100 patches, $P = 100$, leading to a feature vector of 3,200 bits for each document.

3.2.3 Creating the document fingerprint

From the feature vector we can create a document fingerprint that can be used to authenticate the document without revealing information about the document. The fingerprinting method should be both concise and robust to errors. This situation is similar to that of biometrics, where a user provides a value $\widetilde{\mathbf{V}}$ which is close to, but not identical to the registered value $\mathbf{V}$ (e.g., the Hamming distance is relatively small). Additionally, providing an adversary with the full feature vector may not be desirable, as it provides a blueprint for potential forgery.

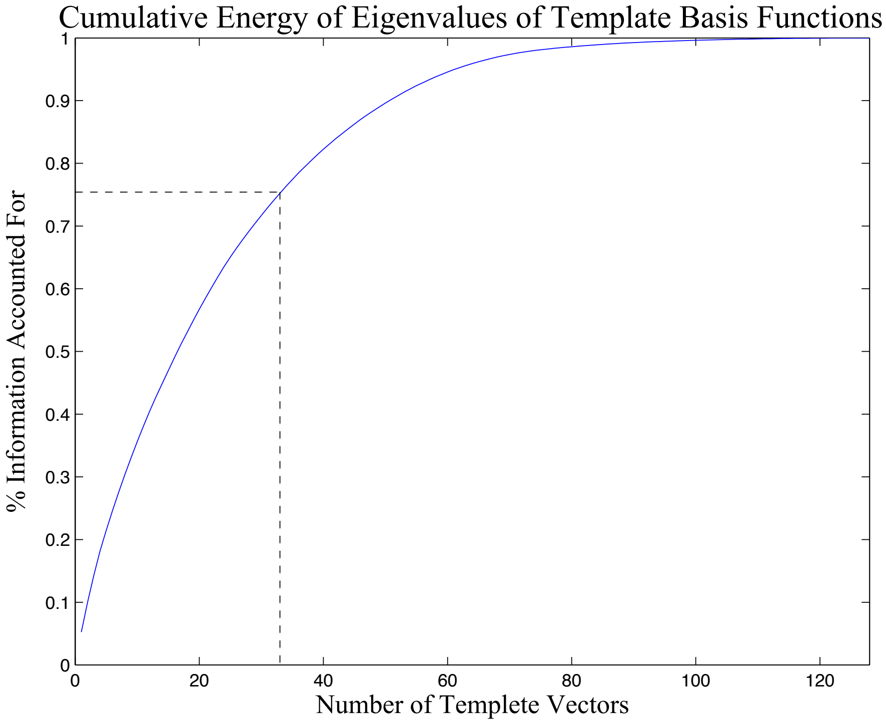


Figure 3.9: Principal component analysis of a large number of 8×8-patches shows that 75% of the information has been extracted after 32 components.

A document fingerprint consists of a hash of the feature vector $H(\mathbf{V})$, where H is a collision-resistant cryptographic hash function, along with a secure sketch $ss(\mathbf{V})$ following the ideas of Dodis et al. [32] and Juels and Wattenberg [53]. The secure sketch allows the system to correct any errors that may occur in the candidate $\widetilde{\mathbf{V}}$, assuming $\widetilde{\mathbf{V}}$ is close enough to $\mathbf{V}$, without revealing $\mathbf{V}$ to an adversary who does not know any $\widetilde{\mathbf{V}}$ close to $\mathbf{V}$.

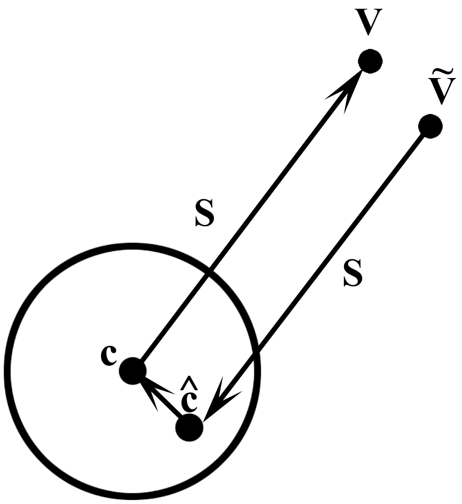


Figure 3.10: The secure sketch of vector $\mathbf{V}$ is mapped to a random codeword c by choosing a random offset S. The secure sketch of $\mathbf{V}$ consists only of this offset, S, and its cryptographic hash, $H(\mathbf{V})$. Verifying the identify of a document based on its purported feature vector $\widetilde{\mathbf{V}}$, which is close to $\mathbf{V}$, consists of three steps. First, shift $\widetilde{\mathbf{V}}$ by S to $\hat{c}$. If $\widetilde{\mathbf{V}}$ and $\mathbf{V}$ are close, then so are c and $\hat{c}$. Second, correct the corrupted codeword $\hat{c}$ to the nearest valid codeword, c. Finally, shift c by S, and compare its hash with $H(\mathbf{V})$.

Suppose the registered value for a document is an N-bit value $\mathbf{V}$, and we wish to accept any $\widetilde{\mathbf{V}}$ within Hamming distance δN of $\mathbf{V}$. The secure sketch proposed by Juels and Wattenberg chooses a random codeword c from an error-correcting code of length N that can correct δN errors, and stores $ss(\mathbf{V}) = \mathbf{V} \oplus c$. This can be viewed of as a shift, S, from $\mathbf{V}$ to c. To recover $\mathbf{V}$ from a candidate $\widetilde{\mathbf{V}}$, the system calculates $\hat{c} = ss(\mathbf{V}) \oplus \widetilde{\mathbf{V}}$. This can be thought of as a shift from $\widetilde{\mathbf{V}}$ to $\hat{c}$. If the distance between

$\mathbf{V}$ and $\widetilde{\mathbf{V}}$ is greater than δN, then the codeword $\hat{c}$ will not be close enough to the original codeword c, i.e. within δN of c. The next step corrects $\hat{c}$ to the nearest valid codeword c, and verifies that $H(\mathbf{V}) = H(c \oplus ss(\mathbf{V}))$. Only if $\mathbf{V}$ and $\widetilde{\mathbf{V}}$ have Hamming distance less than δN will the system correctly output $\mathbf{V}$. See Figure 3.10 for a two-dimensional representation of the recovery of $\mathbf{V}$ from a candidate feature vector $\widetilde{\mathbf{V}}$.

Dodis et al. [32] show that the number of bits of information about the feature vector revealed by the secure sketch is $N - k$, where $k = \log K$ is the dimension of the error-correcting code used in the secure sketch when it has K codewords. Thus, in order to maximize the security of the system for a fixed N, the error-correcting code should have as high a dimension k as possible.

Low-Density Parity Check (LDPC) codes, along with turbo codes, are among the strongest error-correcting codes in use, thanks to efficient decoding algorithms developed in the last two decades. In our implementation, we used the LDPC library written by Neal [72]. LDPC codes are well-suited to this application because they work well on large block sizes [62].[2] In addition, the LDPC decoding algorithm can take into account a confidence level specified for each individual bit to further improve the performance of the code. In our case, this confidence level can be calculated from the magnitude of the dot product of the template vector with the patch vector. The correspondence between the two is graphed in Figure 3.12.

The length of our feature vector is $N = 3200$ bits. We experimented with codes of suitable dimension to correct bit error rates between $\delta = 10\%$, allowing correct identification of all types of paper we experimented with under ideal conditions (see Figure 3.13), and $\delta = 30\%$, suitable to identify documents under less-ideal conditions such as soaking and scribbling (see Figure 3.14). For the case of decoding under ideal

[2] In practice, LDPC codes can often decode beyond the minimum distance of the code. This could allow an adversary to gain an advantage. If an application allows for probabilistic decoding, block sizes can be chosen such that the practical impact is minimized.

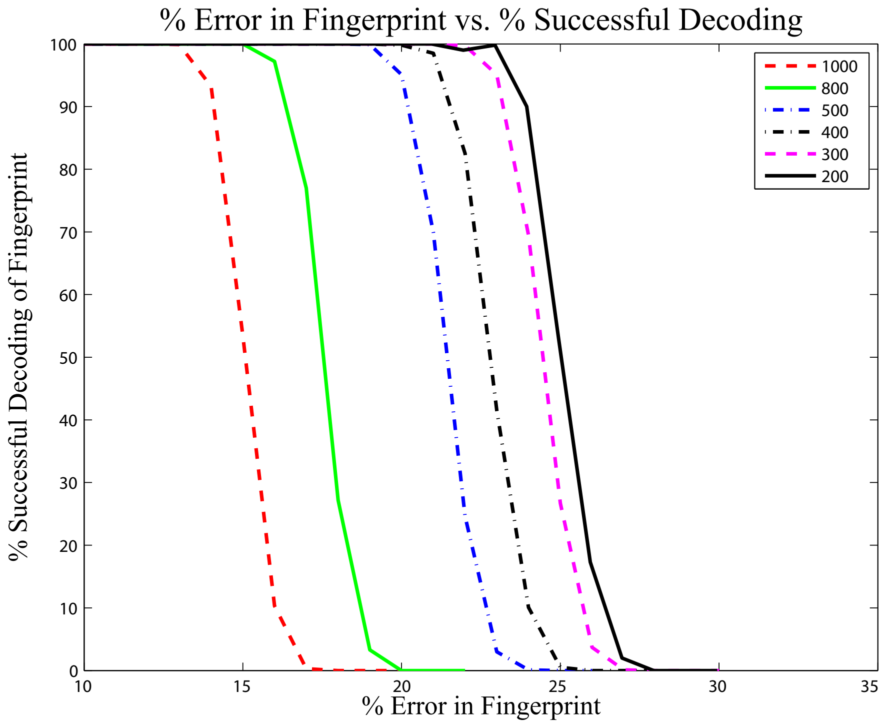


Figure 3.11: Fraction of fingerprints successfully decoded for varying fingerprint error rates using LDPC codes of different dimensions k.

conditions, a code of dimension $k = 1000$ and $N = 3200$ is sufficient to correctly verify all test documents, with no false positives. For the case of decoding under less ideal conditions, a code of dimension $k = 300$ and $N = 3200$ sufficed to correctly verify 95% of all test documents, with no false positives. See Figure 3.11 for a summary of these results.

The feature vector length can be adjusted to suit the needs of the application (and the expected document treatment conditions) by increasing or reducing the number of patches. Longer feature vectors provide a higher level of accuracy when distinguishing two documents, especially under harsh treatment, but require increased storage. We chose $N = 3200$ bits as our feature vector length to ensure that it would fit in a 2-D barcode.

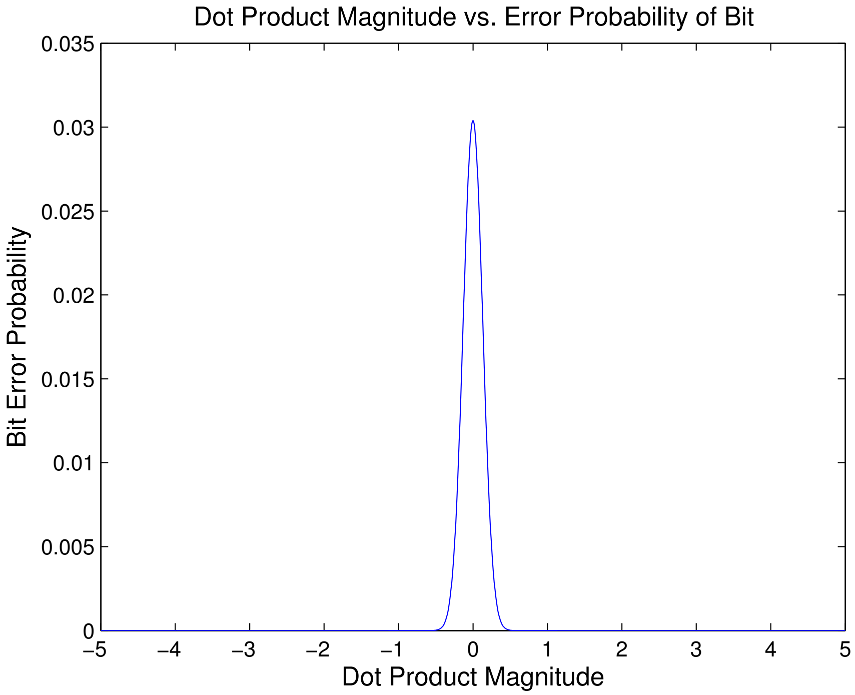


Figure 3.12: Correspondence between dot product magnitude and error probability during validation. (Fit to normal distribution with μ=0 and σ=0.1314.)

3.3 Robustness

Section 3.2 describes a process for registration and validation of a document fingerprint. In this section we evaluate document fingerprints across different types of paper, including normal copy paper (Boise Aspen 50), university letterhead with a visible watermark, and index cards. We also evaluate the fragility of a document fingerprint under various treatment conditions. Our goal is to test whether our technique typically validates different observations of the same document (true positives) and rejects pairs of observations of different documents (true negatives), while rarely validating pairs of observations of different documents (false positives) or rejecting different observations of the same document (false negatives).

Our experiments show that a fingerprint can be found for a variety of different types of paper. Unless otherwise noted, each experiment began with a document scanned at 1200 DPI on an Epson Perfection v700 scanner. Each test focused on a 3×3 inch square in the center of the page.[3]

For each test, we captured five observations of a set of five documents, for a total of 25 observations. Each observation consisted of four scans taken at 0°, 90°, 180°, and 270° that we used to estimate surface normals. We expect no two scans of the same document to be exactly alike due to slight variations in placement on the scanner, random noise, and other uncontrolled variables in the process.

The amount of error tolerated in a matching fingerprint can be adjusted by choosing an appropriate error-correcting code during the fingerprinting process described in Section 3.2. The number of bits that can be corrected by the code should be determined by the needs of the application, as it establishes a tradeoff between the security of the system and the relative likelihood (or harm) of a false positive or false negative.

3.3.1 Ideal handling conditions

As a baseline test, we measured the frequency of correct and incorrect validation and rejection under ideal handling conditions, when we expect no document degradation.

We began with 25 observations (5 from each of 5 documents). We chose 40 random seeds and sampled a fingerprint from each observation for each seed, following the process described in Section 3.2. We compared each of the $\binom{25}{2} = 300$ pairs of observations for each seed, yielding a total of $12,000$ comparisons.

For each comparison, we computed the Hamming distance between the two fingerprints. These distances are summarized for the 12,000 comparisons by the histogram shown in the top graph in Figure 3.13. Under non-adversarial conditions, document

[3] In our robustness experiments, we used a printed box on the test pages to identify the region to be fingerprinted and to align the different scans. However, alignment could be accomplished by other means, such as by relying on the boundaries of the page or other printed material, or by simply recording a few patches at high resolution [89]. Different sets of patches should be used for alignment and verification, because using the same patches could increase the false positive rate.

fingerprints of the same piece of normal copy paper differ, on average, in only 99 (3.1%) of the 3200 bits. In contrast, as one would expect, the average Hamming distance for fingerprints made from observations of *different* documents is 50% of the bits. These distributions are well-separated; the maximum Hamming distance between feature vectors from the same document is 177, while the minimum distance between feature vectors of different documents is 1490. An error tolerance anywhere in this range would give no false positives and no false negatives for these tests. In practice, the preferred error tolerance will be closer to the midpoint, allowing for maximum flexibility to accept or reject a candidate document. We repeated this experiment on index cards and university letterhead, producing similar results. see Figure 3.13.

The distributions for the "same" fingerprint comparison tests and "different" fingerprint comparison tests seem to be reasonably approximated by a normal distribution. Fitting Gaussian curves to this data, we can find a summary statistic, Egan's *sensitivity index* for Gaussian distributions of signal and noise with unequal variances, given by:

$$d_s = 2(\mu_2 - \mu_1)/(\sigma_1 + \sigma_2) \tag{3.4}$$

where μ_1, μ_2, σ_1 and σ_2 are the means and standard deviations of the distributions [66]. For this experiment, $d_s = 52.0$. To give some intuition about the significance of this statistic, the two Gaussians intersect at a Hamming distance of 731 bits; the heights of the curves are such that the chance of a single comparison resulting in either a false positive or false negative is 1 in 10^{148}. If we reduce the feature vector length from $N = 3200$ bits to 1600, 800, or 400 bits, the probability of such errors is 1 in 10^{96}, 1 in 10^{57}, or 1 in 10^{35}, respectively.

We repeated these experiments on different scanner models and found similar results. When comparing a document fingerprinted on one model and verified on another, results are slightly worse.[4] This experiment demonstrates that individual sheets of paper can be re-identified under ideal conditions with high accuracy. In many applications, it is necessary to be able to identify a sheet of paper even when it is subjected to harsh treatment. The next section evaluates the ability to identify each sheet of paper after it is treated harshly.

3.3.2 Non-ideal handling conditions

The previous experiments were performed under ideal handling conditions. We performed additional tests to ascertain the robustness of fingerprints when the document is subjected to less-than-ideal conditions. These tests included scribbling on the paper, printing on it with ink, and soaking the page in water.

Scribbling

We first scanned a set of five blank documents, then scanned them again after scribbling over them with a pen. In each document the scribble was unique, covering an average of 8% of the test region. In this test, 25 pre-scribble observations were compared against their 25 post-scribble counterparts, for a total of 625 pairs. We used 40 different fingerprint samples per document to yield a total of 25,000 comparisons. The Hamming distances resulting from these comparisons are plotted in the graph in Figure 3.14. The sensitivity index in this case is lower ($d_s = 28.8$), although the curves remain quite well-separated. With a decision threshold of 1130 bit errors in the fingerprint, the chance of a false positive or false negative is 1 in 10^{47}.

[4] We chose several of the parameters of our algorithm (e.g., the downsample factor and size of the patches) based on preliminary experiments using the Epson v700 scanner. The optimal settings for verification of documents using other scanner models may vary.

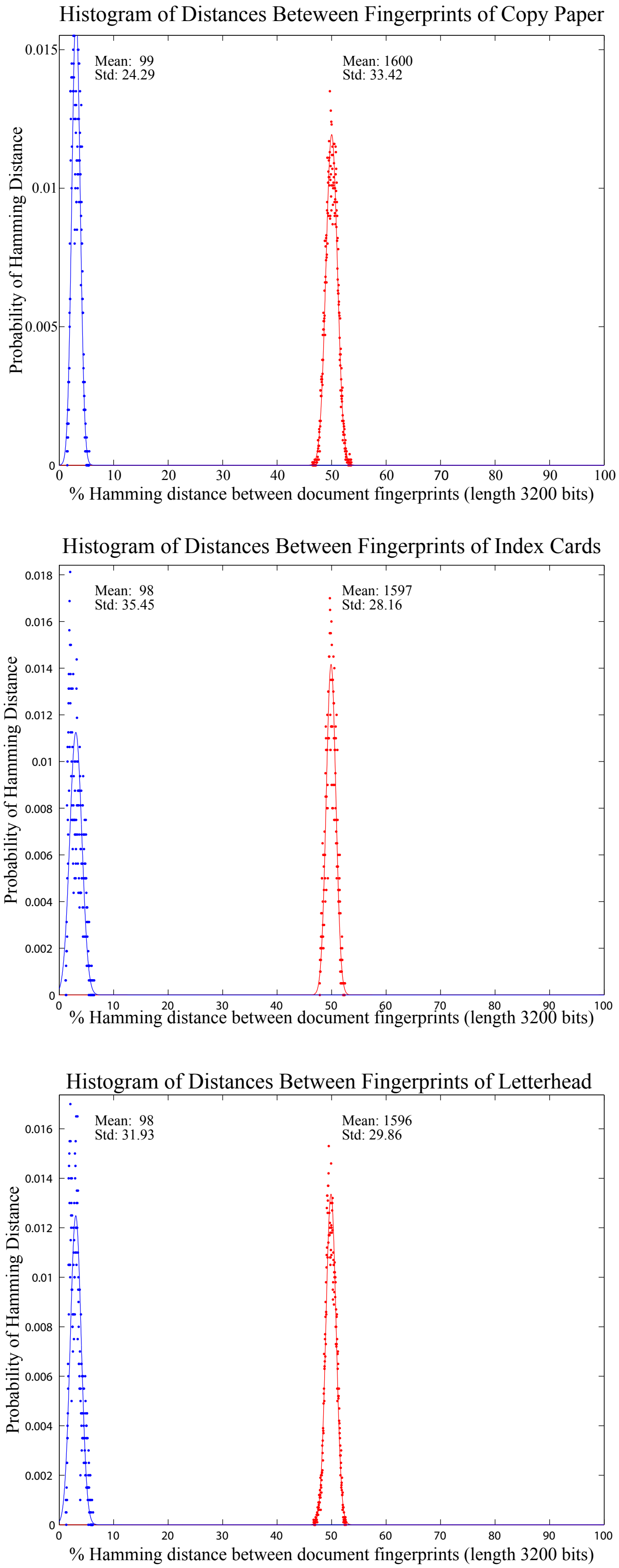
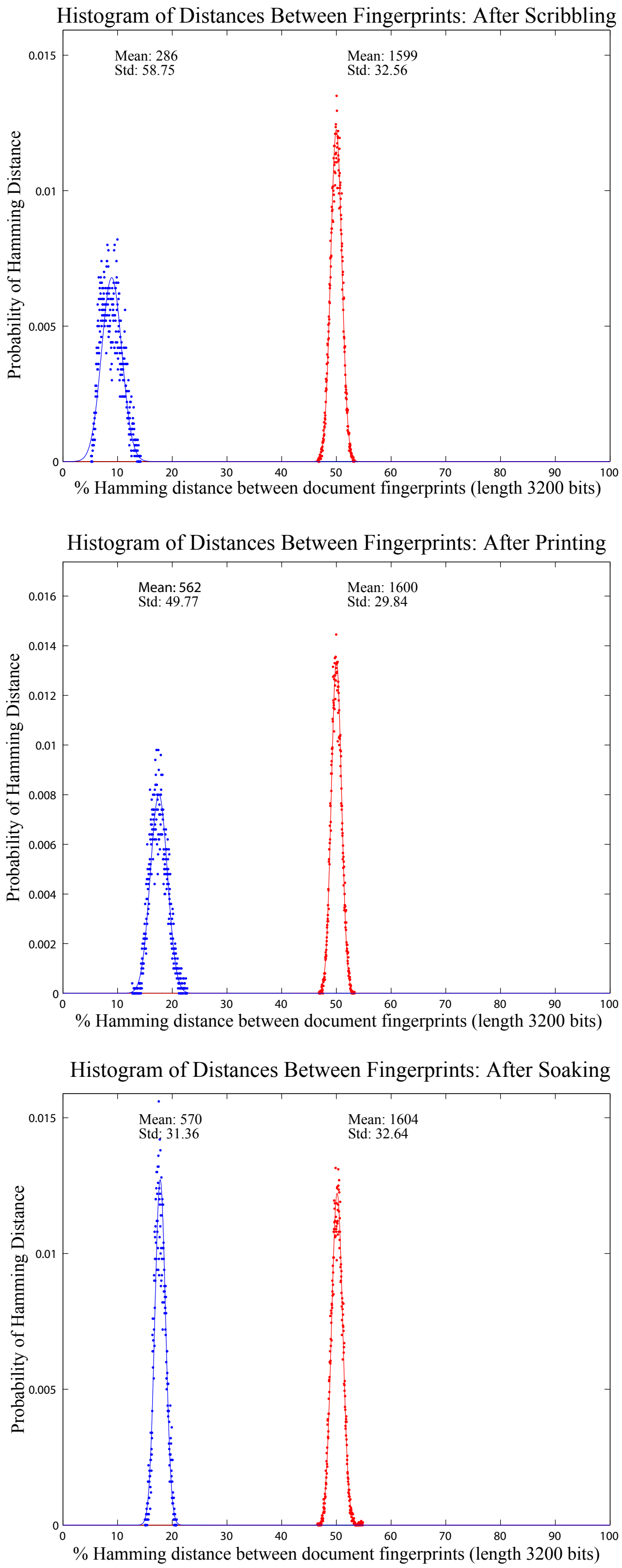


Figure 3.13: Distributions of Hamming distances between fingerprints for three paper types: copy paper *(top)*, index cards *(middle)*, and letterhead *(bottom)*. In all graphs, the curve on the left depicts the distribution for scans of the same document, while the curve on the right gives the distribution for different documents The curves remain well separated even under these adverse conditions.


Histogram of Distances Between Fingerprints: After Scribbling

Histogram of Distances Between Fingerprints: After Printing

Histogram of Distances Between Fingerprints: After Soaking

Figure 3.14: Distributions of Hamming distances after subjecting documents to non-ideal treatments: scribbling *(top)*, printing *(middle)*, and soaking in water *(bottom)*

Printing

In this experiment we printed single-spaced text in 12 pt. Times New Roman lettering over the test region, covering approximately 13% of the area with ink. The distributions shown in the middle graph of Figure 3.14 were obtained as in the scribble test. Even in this experiment, in which most patches used for the fingerprint were partially covered by ink, the sensitivity index is 26.1 and the chance of a false positive or false negative at the crossover is 1 in 10^{38}.

Wetting and drying

The bottom graph in Figure 3.14 shows the resiliency of document fingerprints after the document was submerged in water for five minutes. We dried each test document and ironed it until it was as flat as possible. Using the same evaluation protocol as for the scribble and printing tests, we found that documents could still be validated with 100% reliability, even with this fairly extreme mistreatment of the page ($d_s = 33.3$).

These experiments demonstrate that our fingerprinting method is robust when a document is handled under certain rough conditions. The ability to identify a document before and after it is printed on, scribbled on, or soaked in water has many potential applications.

3.4 Security

The security of our method relies on the inability of an attacker to reproduce the document's surface, either because he does not know what features to produce or because he cannot recreate the normal vectors at the required scale. The threat model of each application is determined by several factors: the availability of an original to the attacker, whether verification is performed online or offline, and whether the verification device is trusted. Under the most common threat models, our method should prevent an attacker from forging a copy of an original.

Performing verification online or offline results in different considerations. Here "online" means that the verification device can communicate with a remote trusted server which can store data and perform computations; "offline" means that the verification device cannot communicate, although it can be preprogrammed with a limited amount of information such as a constant number of cryptographic keys. Online verification of a document has a straightforward solution, while offline verification requires security tradeoffs.

3.4.1 Online verification

Online verification need not reveal in advance the patch locations that will be analyzed. This forces an attacker to reproduce the entire surface of a document before presenting it for verification. In one approach, the verification server requests complete raw scans of the document at each of four orientations, which the server uses to perform the verification algorithm. Under this construction, the verification server does not reveal the chosen patches.

In an alternative approach, the verification server provides a fresh pseudorandom challenge to the client, and the client uses the challenge to seed a pseudorandom generator which is used to pick the patches and templates used in the verification algorithm. The client then computes the feature vector and sends it to the server. The server, having computed the same feature vector on its stored scans of the original document, verifies that the two feature vectors are similar enough.

In this threat model an attacker does not know *a priori* which patch locations on a document will be sampled. This forces an attacker to reproduce the surface texture of the document at each sample point in order to pass a counterfeit as an original.

3.4.2 Offline verification

The security of offline verification depends on whether the verification client is trusted and on the availability of an original to the attacker. In the offline case, we assume that the fingerprint of the legitimate original document is either pre-stored on the client or is printed onto the document (perhaps as a 2-D barcode) along with the authority's digital signature of the fingerprint. In either case, the client device checks the document against a known fingerprint.

Offline: trusted device

Currency and ticket counterfeit detection at banks and concerts are two important examples of offline verification with a trusted device. By "trusted" we mean that the device outputs a Boolean match/no-match result but does not leak any other information.

The secret information stored in the device could be the public key of the registration entity. The seed stored in the document fingerprint could be encrypted under the private key of the registration entity.[5] Therefore, knowledge of the fingerprint for a document does not reveal the patch locations. The hash of the feature vector could also be signed by the registration entity, preferably using a separate key. This allows only trusted devices to determine patch locations and verify the authenticity of a document. No access to the registration entity is required, provided that the device has knowledge of the decryption and verification keys of the registration entity.

In this threat model the adversary does not know which patches will be analyzed. This forces the attacker to recreate the surface normals across the entire document to ensure verification of the document.

[5] In this scenario we treat the public key *and* private key as secret. Storing the public key on the device prevents a successful attacker from generating valid fingerprints and can only reveal which patch locations will be sampled.

Offline: untrusted device, no access to original

In the next case, the verification device is offline and untrusted (i.e., it might leak everything it knows to the attacker) and the attacker has not seen the original document that he is trying to forge. In this case, the attacker cannot forge the document because he does not know anything useful about the normal field that he must create. At most, he knows the fingerprint (if it is stored in the device) but this does not help him because the fingerprint is a secure sketch.

Offline: untrusted device, access to original

The final case is the most challenging one, where the verification device is offline and untrusted, and the attacker has access to an original document that he wants to copy. Because there are no secrets from the attacker—he sees the original document including anything printed on it, and he knows the full state of the device—the attacker knows exactly which patches will be used for verification and how the feature vector will be computed from those patches. The attacker's only task is to make a document that will generate a feature vector close enough to the original. Whether the attacker can do this is the subject of the next section.

3.5 Forging a Document

Suppose an attacker has seen an original document and wants to create a second document that will pass verification as the original. The attacker will start with an arbitrary piece of paper which, by assumption, will have a very different feature vector from the original. He will then try to modify the target paper so that its feature vector is close to that of the original document. To do this, the attacker needs to make fine-grained modifications to the document's surface normals. This might be done via lithography or photographic techniques, but these will be expensive and will probably require special equipment. The most effective, economical way to control the surface, we believe, is to print on the document.

Equation (3.2) shows that the fingerprinted vector (d_x, d_y) contains additional factors s and ρ. The scanner-dependent factor s can be assumed to be fairly constant across the page and hence has no influence on the sign of the correlation results in the feature vector generation. The remaining scale is given by the local surface reflectance ρ of the paper at a given location, which should be stable across multiple scans. On empty paper it is nearly constant; in the presence of print, ρ is greatly attenuated, which lessens the influence of the printed portion onto the correlation result. The adversary can try to control bits of the feature vector by printing dark ink at selected points in order to reduce their influence in the correlation calculations. Besides reducing ρ, printing at a point tends to flatten the document surface, as shown in Figure 3.6.

An adversary who aims at forging a document might try to leverage these effects by printing a set of carefully placed dots, either to cause the surface texture of a candidate document to express the same fingerprint as an original, or to down-weight unfavorable contributions to the patch correlation. To do this the forger must overcome two hurdles: printing dots on the page at desirable locations and/or printing dots with favorable surface normal vectors. Dark ink on a document would directly affect reflectivity, while light ink might solely change the normal vectors at a specific location. We assume that the adversary uses commercially available equipment and is able to print dots in any color from black to white. He has less control over the exact shape of the dots, which varies by printing technology and type of paper.

We conducted experiments to characterize the ability of a forger to precisely control the effect of a printed pattern. We measured the effective resolution—the number of distinct printable dots—for a high-end office printer, a Xerox Phaser 8550, with a nominal resolution of 2400x2400 DPI. The effective resolution is limited by dot gain,

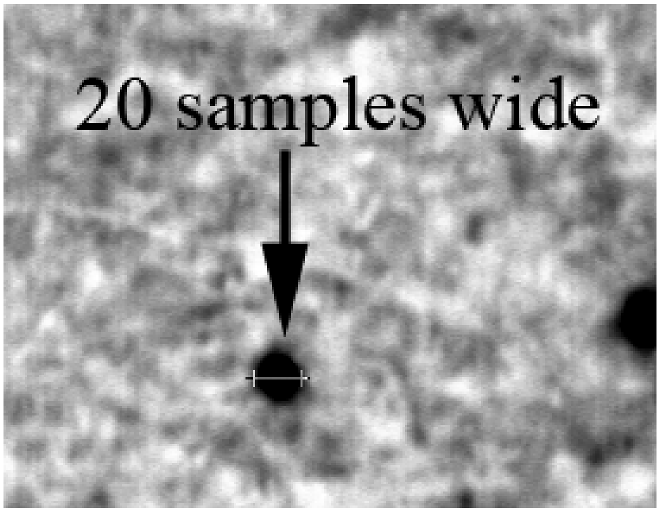


Figure 3.15: The smallest dots that can be produced by our test printer are 1/240 inch—20 samples wide in this 4800 SPI scan—despite the printer's nominal 2400 DPI positional accuracy.

which causes printed dots to become larger than intended (see Figure 3.15) due to factors such as the viscosity of the ink and the absorbency of the paper. The smallest dots that the test printer could produce on a normal piece of copy paper are 1/240 inch, or 20x20 samples when scanned at 4800 SPI. This limits the effective resolution to 240 DPI. On the other hand, the positional accuracy of the printer seems closer to the rated 2400 DPI. We conclude that a forger could use commodity printers to print dots with positional accuracy similar to what commodity scanners can measure but size much greater than the scanner's pixels.

Because printed dots typically span more than one sample in a patch, printing a dot at a specific location affects the neighboring surface normal vectors in unpredictable and uncontrollable ways. Due to paper variations as well as limited precision in the placement and viscosity of ink, the forger does not have precise control over the normal vectors caused by a dot. We performed an experiment where we printed a series of black dots in a region of the document. We identified the black dots and measured the normal vectors in the surrounding region. For each printed dot, the desired normal vector of a location occurred on only one point of the surface.

The bottom-line question is how many degrees of freedom the adversary has in controllably modifying the normal vector field in a patch. Given the linear transformation used to determine each feature vector bit, the adversary will likely be able to achieve a desired set of feature vector values if he has enough degrees of freedom.

If there are N feature vector bits, and each bit is computed as the sign of the correlation of the normal field with a random vector field, then a truly random normal field value would match all N feature vector bits with probability 2^{-N}. However, it is likely that the feature vector bits are not fully independent. Although we have some evidence about the degree of independence (see, e.g., Figure 3.9), we do not have a precise estimate of how much entropy is encoded in the feature vector.

We are thus left with an open question: does the amount of information in a patch, as encoded in a feature vector, exceed the adversary's ability to inject information into the patch? If we knew the answer to this question, we could state with some confidence whether an adversary could forge a document in the most favorable case (for the adversary), where the adversary sees the original document and the verification device is offline and untrusted. Unfortunately, we have to leave this question for future work.

3.6 Applications

Many applications could benefit from the ability to uniquely identify a document. Many situations where physical possession of an item must be verified or authentication of an item is required could fruitfully employ our technique. Currency, ticket, and art counterfeit detection, as well as verification of product packaging are some of the applications where physical document authentication is desirable.

Counterfeit currency detection is one obvious application. The financial impact of counterfeit currency is large. Estimates of annual global revenue loss range from \$250- to \$500 billion [6, 98]. The ability to authenticate bills could change the way currency is produced and fraud is detected. Such a system would begin during currency production. The government would generate a fingerprint for each bill. This fingerprint could be stored in a database, along with the bill's serial number, or the government could digitally sign the fingerprint and print the fingerprint and signature on the bill. Any party wishing to verify a particular bill would scan the bill and verify that the fingerprint matched the one signed by the government. The authentication of a bill could be performed offline or online. Businesses and banks accepting large cash deposits could verify the currency was legitimate before completing the transaction. Offline authentication could be performed provided that the verification device had the public key of the currency issuer.

Ticket forgery at major concerts and sporting events is another large black-market business. Counterfeit event passes were widespread at the 2008 Beijing Olympics [11], and a British website recently sold more than $2.5 million in fake tickets [96]. The ability for purchasers to verify the authenticity of tickets prior to purchase could greatly reduce the prevalence of online ticket fraud. Trust in ticket purchases on websites such as Stub Hub and eBay could be dramatically increased if the seller had to prove access to the item being auctioned or sold. Ticket clearing houses such as Ticketmaster could maintain an online database of fingerprints for all purchased tickets. Any party selling a ticket could scan and upload the ticket to Ticketmaster and receive verification of authenticity.

Forgery of artwork is a black-market business where the application of our technique may not be initially obvious. European police estimate that over half of the works in international markets are forgeries [86]. One family of art forgers was able to make $2 million before they were caught. The ability of art forgers to reproduce the individual brush strokes of a work makes authenticating paintings increasingly difficult. In the best forgeries, art verifiers must sometimes rely on the chain of custody of the work in order to authenticate it [90]. However, we believe that it would be difficult to duplicate features of the *canvas* (down to the detailed arrangement of the weave) upon which the work is painted. Thus art authenticity or forgery might be detectable by applying a technique like ours to the canvas, most probably on the back side of the painting.

Lottery tickets are similar to currency except that players need not be aware of a fingerprinting technique at all. In order for a lottery winner to collect on their winnings, the ticket must be verified by the lottery authority. The fingerprint of a winning ticket need not be printed on the document at all. Fingerprints of all possible winning lottery tickets can be privately maintained, and any claimants can be required to produce the actual winning ticket, with correctly verified fingerprint, in order to collect their winnings.

The accurate identification of paper based product packaging could benefit from this technique as well. When inspecting cargo, customs officials often inspect the contents of packages to weed out counterfeit goods. We can increase confidence in package contents by authenticating a product's packaging. If the packaging of a product is legitimate, then the contents of the package have a much higher likelihood of being authentic.

3.7 Privacy Implications

The feasibility of paper-based authentication can have positive or negative implications depending on the application under study. Because our results do not modify the paper in any way, there is no way to detect, by inspecting a piece of paper, whether its fingerprint might have been recorded in advance by an adversary. This fact violates the traditional assumption that pieces of paper cannot easily be traced without the addition of distinguishing marks. Even unopened sheaves of blank printer paper might in principle have been fingerprinted at the factory. This result has important implications for a number of applications, particularly paper-based voting. Chapter 5 discusses the implications of paper fingerprinting for paper-based elections, developing an efficient election auditing technique as well as a new method to detect ballot box stuffing. Each of these techniques take advantage of the ability to re-identify individual sheets of paper.

More generally, the ability to re-identify ordinary sheets of paper casts doubt on any purportedly private information gathering process that relies on paper forms. "Anonymous" surveys or reporting systems may not in fact be anonymous. Though it has long been possible to track sheets of paper using subtle chemical markers or "invisible ink," these methods require some level of special expertise, and the presence of markers leaves evidence of the attack. Our research shows that an attacker armed with only ordinary equipment—a commodity scanner—is able to re-identify paper reliably without leaving any telltale marks.

3.8 Discussion and Potential Future Work

This chapter described how ordinary pieces of paper can be fingerprinted and later identified using commodity desktop scanners. The technique we developed functions like a "biometric" for paper and allows original documents to be securely and reliably distinguished from copies or forgeries.

At least two questions remain to be answered in future work. First, in the threat model where the adversary has access to the original document and the fingerprint, we do not know for certain that a clever adversary cannot forge a copy of the document with a high-resolution printer. Our initial work could not determine conclusively whether an adversary who can use a good printer will have enough degrees of freedom in modifying a document to make the document match a known fingerprint. Second, while we conjecture that our method can be applied to other materials such as fabric, more testing is needed to verify this, and special methods might be needed for some materials. We leave both of these questions for future work.

Our results are a tribute to the resolution of today's scanners. Future scanners will capture ever more detailed images of paper documents, eventually allowing individual wood fibers to be imaged clearly. The security of our methods against forgery, in cases where the adversary has full access to information, will depend ultimately on a race between the resolution of the printers that can create patterns on a page, and the resolution of the scanners that can observe patterns.

3.9 Alternate Approaches

Section 3.2 introduces a process for fingerprinting and verifying the fingerprint of a document. In this appendix we briefly outline some alternative strategies that might be desirable under different criteria for robustness or different levels of concern about forgery.

Using albedo versus normals

Because the high-resolution paper scans shown in figures throughout this chapter reveal obvious color variation in addition to surface texture, perhaps a more straightforward approach would be to use the albedo (color) of the page as the basis for a fingerprint, rather than, or in addition to, the shape. Indeed, our initial implementations explored this approach, using a single scan (which combines albedo and normal information) to construct the fingerprint of a document. This approach is simpler and offers the substantial benefit that the document can be fingerprinted or verified more quickly, through a single scan.

The intensities of most of the pixels in a scanned page are modeled well by a truncated normal distribution, centered around the "white" color. To use this data as the basis for a fingerprint, we simply construct the vector $\mathbf{p}$ as the concatenation of these intensities from a given patch. For example, an 8×8 patch would yield a vector $\mathbf{p} \in \mathbb{R}^{64}$. The fingerprint is then extracted from a collection of patches as described in Algorithm 1.

We did not pursue this approach because we believe this form of fingerprint may not resist forgers who use very light ink to print a desired pattern on the page. Another drawback is that any black ink on the page, which lies well outside the roughly-normal distribution of intensities found in blank paper, contributes to a very strong negative value in $\mathbf{p}$, introducing a bias in the dot products for the patch. Thus, any value outside the range of the truncated normal distribution must be zeroed out before constructing the fingerprint. This provides another opportunity for a forger to deliberately zero out regions of the patch with the goal of flipping bits towards a desired fingerprint. These attacks might be difficult to carry out in practice, since they require excellent registration in the printing process. Therefore, albedo-based fingerprints may be suitable for applications where some added risk of forgery is an acceptable tradeoff for increased speed and simplicity.

Patch-pair comparisons

Recall from Algorithm 1 that the vector $\mathbf{p}$ contributes K bits to the overall fingerprint by taking the signs of the dot product of $\mathbf{p}$ and a series of ortho-normal template vectors. We have also considered (and implemented) an alternate version of the algorithm where the bits of the fingerprint are taken to be the signs of the dot products of pairs of patches $\mathbf{p}$ and $\mathbf{q}$. The naïve version divides the pool of patch positions into pairs and computes one bit of the feature vector from each pair. Unfortunately that approach allows an attacker to tweak each pair in turn independently. A more robust version considers bits from *all* patch pairs (p, q) where $p \neq q$. For example, for

64 patches each patch would participate in 63 bits, and this scheme could generate $\binom{64}{2} = 2016$ total bits.

In the case where a forger has a copy of an original document and therefore knows the fingerprint he is trying to reproduce (Section 3.4.2), this formulation has the advantage that the bits of the fingerprint are more tightly bound than those of the template vectors. Any attack on a single bit—for example, printing on a patch—is likely to impinge on the other (62) bits affected by that patch. Thus, a forger would have to solve an optimization problem to figure out how best to perform the attack.

However, the bits of the fingerprint generated from all patch pairs seem to be less independent than the bits generated by the template vectors. Preliminary experiments similar to those described in Section 3.3 indicate that "all-pairs" bits are *mostly* independent, but not as independent as the "template" bits. Since the arguments in Section 3.4.2 for security against "blind" attackers rely on bit independence, we generally prefer the "template" scheme.

Short fingerprints with no error-correcting information

Section 3.2.2 describes a process for generating fingerprints composed of a hash of 3200 or more bits concatenated with some error correction bits. For some applications requiring less security, fewer feature vector bits may be used. Suppose only 100 bits are used, and further suppose that the application tends to produce fewer bit errors (say 15% or less). In such scenarios an alternate approach would be to simply record the hash of those bits. An attacker, without the benefit of the original, is forced to guess among 2^{100} bit sequences, checking guesses against the hash. Unfortunately, this leaves the naïve authentication process with no way to do error correction other than to guess among the roughly 10^{17} strings within Hamming distance of 15 of the sequence extracted from a page—easier, but also daunting.

Fortunately, there is a better approach for the authentication process. Recall that the bits of the fingerprint are taken as the signs of a series of dot products (patches and templates). We have observed that these dot products are well-modeled by samples from a truncated normal distribution. Moreover, we have also observed that the flipped bits mostly come from dot products near zero, and that the bit-flipping process seems to be well-modeled by the addition of "noise" also selected from a truncated normal distribution (with smaller standard deviation than that of the "signal"). $erf()$. With this model in hand, the verification process can search for bit strings similar to the extracted fingerprint while determines taking into account which bits are more likely to have flipped. Specifically, the process repeatedly chooses a bit at random, and flips its value with a probability relative to the likelihood that the chosen bit flipped, each time checking the resulting fingerprint against the secure hash. We simulated this approach and found that about 90% of the time it will find the correct string within 10^6 guesses for the example distribution described above.

The benefits of this approach are that it is simple to implement and provides *no* information to an attacker in the form of error-correction bits. The main disadvantages are that it does not scale well to longer bit sequences and that the stochastic nature of the algorithm provides only probabilistic guarantees of running time. Therefore, it would likely be used only in conjunction with other approaches. For example, an application might attempt this method for offline verification and fall back to an online method in cases when it fails.

Chapter 4

Some Consequences of Paper Fingerprinting for Elections

Elections are critical to the proper functioning of democratic society. As such, there are a number of attributes that must be satisfied for citizenry to trust election results. Elections must be fair, transparent and their results verifiable. At times these attributes seem to conflict with one another. Yet, voting systems exist which attempt to satisfy each attribute. In most cases, a voter's identity is verified prior to their casting a ballot, and they are only allowed to cast a single ballot. Additionally, a voter's particular ballot selections must not be revealed. This property is called the secret ballot. After an election, all cast ballots must be counted accurately and transparently, with the results reported in a timely manner. Accomplishing these tasks seems, at first, to be straightforward, but there are myriad examples where subtle flaws can exist.

The attribute that most complicates accurate and fair elections is maintaining the secret ballot. The secret ballot is intended to prevent voters from intentionally, or unintentionally, revealing their selections. If an individual voter could prove the way they voted, this would lead to an obvious method of voter intimidation or coercion, undermining voter confidence in the fairness of elections.

Historically, voting systems utilized paper as the medium for recording voter's selections. While the past decade has seen the development of a number of, alternative, electronic-based voting systems, most jurisdictions still rely on paper ballots as the medium of record for recording a voter's selections. These voting systems also consist of procedures designed to prevent, or at least detect, election fraud, often through complex auditing schemes. The majority of these voting systems did not consider the ability to fingerprint blank sheets of paper in their designs. In this chapter, we focus on the implications of the ability to fingerprint blank sheets of paper, both in order to preserve the secret ballot, and to detect various types of election fraud across a number of voting systems.

Paper records play a pivotal role in many voting systems. Paper is cheap, familiar, and reliable; and paper records can be read and written by people and machines. As a result of paper's positive properties, the voting systems most widely recommended by election security experts rely on keeping paper records of each ballot. Chapter 3 introduces one method to fingerprint individual sheets of paper based on their unique surface texture. This chapter focuses on the consequences of this technique, evaluating both the positive and negative implications for elections.

Even when using paper, however, care must be taken to ensure election integrity and ballot secrecy. During an election, paper ballots may be treated harshly. These ballots may be creased or folded, and they are modified by markings indicating voter choices. Any paper identification system must be robust to harsh treatment and moderate levels of marking. Chapter 3 shows that it is possible to re-identify a document even when a sheet of paper is handled harshly or modified by being printed or scribbled on.

In certain scenarios, tracking of individual paper ballots could undermine ballot secrecy, while in others it can enable more efficient auditing techniques. Through vigilance and careful procedures, election officials may mitigate many threats posed by paper fingerprinting while harnessing its benefits.

Traditional paper-based voting systems, optical scan voting systems, and DRE-VVPAT systems[1] result in paper ballots containing the voters' choices for various contests. Suppose that someone has access to the paper ballots (or scans of the ballots) before and after an election. If this person can identify the paper ballot you will use to vote, then when election day concludes, that person can reidentify and recover the paper ballot containing your votes.[2]

Fingerprinting of paper ballots presents additional privacy challenges that must be addressed by election officials to ensure ballot secrecy. Fortunately, many attacks based on fingerprinting are nontrivial; they require access to the paper ballots and certain equipment at particular times. In this chapter, I survey the risks that fingerprinting poses based on details of common voting systems and the malicious party's level of access. Based on common themes, I provide suggestions for election officials to minimize these risks.

The ability to uniquely re-identify paper ballots also has implications for election auditing. When performing an audit, we may select individual ballots for review based on fingerprints, enabling efficient ballot-based auditing. These fingerprints effectively serve as serial numbers on the ballots without posing the same privacy risks as serial numbers. Paper fingerprints may also help uncover attacks that are problematic for some auditing schemes, including attacks that rely on ballot box stuffing. These additional checks can potentially result in greater election integrity.

[1] Direct recording electronic voting machines with a voter-verified (or voter-verifiable) paper trail—a computerized voting system producing redundant paper ballot records that each voter may verify and approve.

[2] Some fingerprinting techniques, including the method described in Chapter 3, can reidentify paper even if the voting process results in markings or creases.

Section 4.1 describes the new threats posed by paper fingerprinting as well as mechanisms that may mitigate these threats. Section 4.2 proposes new auditing techniques that make use of paper fingerprinting. Section 4.3 briefly discusses the impact of paper fingerprinting on an end-to-end voting scheme. Finally, Section 4.4 concludes the discussion.

4.1 New Threats

Although the ability to fingerprint and reidentify paper ballots poses a serious threat to voter privacy, officials may take steps to mitigate these threats. In this section, I detail several threats to voting systems utilizing paper ballots and discuss possible countermeasures.

Even without fingerprinting techniques, a number of methods exist for making paper ballots unique and potentially identifiable. Voters may choose an unusual write-in candidate or a unique combination of candidates to create a distinctive ballot. Given enough races, a pseudorandom selection of choices would with high probability create a distinctive ballot that could be later identified. Alternatively, poll workers may mark ballots with invisible ink or lightly crease the corner of a ballot.

This chapter only considers threats created or made easier by fingerprinting paper. For example, suppose that an optical scan system stores a fingerprint or high resolution scan of every ballot in the order they are cast. Given this information, an observer that watches when voters submit their ballots and can later examine the ballots for fingerprints could reidentify a voter's paper ballot, thus revealing the voters' choices. An easier, equally devastating attack exists without fingerprinting, however: the optical scan machine could simply store all votes cast in order. Therefore, this threat is not considered.

We consider those threats in which someone other than a voter can learn the voter's choices for various contests. This may be with the consent of the voter. To sell her vote, a voter may provide the fingerprint of her paper ballot to a purchaser. If paper ballots are revealed on or after election day, the purchaser can reidentify the appropriate ballot and verify the choices. Alternatively, someone may want the ability to uncover non-consenting voters' choices—whether due to curiosity or a desire to coerce voters—by fingerprinting blank paper ballots in advance and later associating voters with completed ballots. Anyone from voters and election officials to paper mill workers may be a participant in these threat scenarios. The term adversary refers to any party that seeks to undermine ballot secrecy.

In evaluating each of these voting systems, it is important to describe the limits of the adversary. In each example, we are dealing with an adversary who is able to successfully apply the techniques described in Chapter 3 to a large number of blank ballots. This adversary attempts to coerce a voter to make a particular selection. The adversary has the ability to control which ballot his target voter receives, or has the ability to learn, after the election, information about the order in which voters voted, which can reveal information about which ballot the voter used.

This reveals two general threat models relating to paper identification. The first occurs when an adversary is able to fingerprint a significant portion of the ballot stock prior to an election and later associate voters with particular ballots. The second occurs when an individual voter scans her own ballot, either voluntarily or under coercion. I will now discuss threats for various voting systems under each attack model.

4.1.1 Ballot Stock Fingerprinting

In this attack model, an adversary with access to the ballot stock is able to fingerprint or make high-quality scans of a significant portion of the ballots. By combining the ballot fingerprints with information about the order of voters an adversary may be able to undermine ballot secrecy. It is important discuss this attack for various voting systems to gauge the relative impact of this attack on each system.

DRE-VVPAT with Paper Spool. Paper fingerprinting poses a serious threat to DRE-VVPAT systems using printers with paper spools. We do not consider continuous spool-to-spool DRE-VVPAT systems, as they fundamentally fail to protect the secret ballot [55]. In cut-and-drop VVPAT systems, the record tape is used in a fixed order. As each voter casts his ballot, the tape is cut, dropping the segments into a box. Suppose that an adversary has unmonitored access to this paper spool prior to election day. The adversary can unroll the spool and repeatedly fingerprint short segments of the paper tape in order, storing the fingerprints in that order. He can then re-roll the spool and return it. Because the DRE will use the paper tape in order, the order of the segment fingerprints will correspond to the order that ballots are cast on the DRE. Although some segments may be destroyed when the ballots are separated, use of short enough segments can ensure that at least one segment remains intact per paper ballot, allowing complete re-ordering. For this attack to succeed, an adversary would need the ability to scan paper tape segments prior to the election, to observe the order that some or all voters enter the voting booth, and to fingerprint some or all of the resulting paper ballots later.

It is important to note that an adversary need not observe all voters or fingerprint all ballots before and after the election. Suppose that an adversary can reidentify her own ballot without scanning it, perhaps by casting a distinctive ballot, creating a point reference with respect to future ballots. If you enter the voting booth immediately after the adversary, she knows that your ballot will contain the segments immediately following her ballot's segments. Therefore, if the adversary can determine the fingerprint of her ballot after the election, she can guess the fingerprint on your ballot. Numerous similar correlation attacks are possible.

DRE-VVPAT with Paper Sheets. Using a DRE-VVPAT with standard, disconnected paper sheets does not mitigate all threats of a paper spool. Anyone that can fingerprint these sheets and has some knowledge of the order in which they will be loaded can mount attacks similar to those involving paper spools. In this case, an adversary would need the ability to scan some or all paper ballots prior to the election, to gain some knowledge of the order that these paper sheets will be loaded into the DRE, to observe the order that some or all voters enter the voting booth, and to fingerprint some or all of the resulting paper ballots later.

Paper-Based Voting. The issues with paper-based voting are similar to DRE-VVPAT with paper sheets. In this case, a voter may (potentially) have the ability to choose his own blank paper ballot, but the voter may be paid, coerced, or confused into making a nonrandom choice, such as taking the top ballot on the pile. Instead we recommend giving each voter an opportunity to rerandomize the pile of ballots, for example, by cutting the deck. Following this shuffling, an adversary would have greater difficulty inferring a relationship between future voters and their ballot fingerprints. To undermine a paper-based voting system, an adversary would need the ability to scan some or all paper ballots prior to the election, to gain some knowledge of the order of these paper sheets at the poll workers' table, to observe the order that some or all voters receive paper ballots (and, if voters can randomly choose or shuffle the ballots, to draw meaningful inferences in spite of this uncertainty), and to fingerprint some or all of the resulting paper ballots later.

Optical Scan Voting. In many ways, optical scan voting machines present a similar scenario to paper-based voting, but these machines also contain a scanner that may be capable of fingerprinting the ballots that they scan.[3] The optical scan machine may store the fingerprint and votes on each ballot as part of its normal operations and publicly reveal this data later (Section 4.2 describes how this may be helpful for auditing). In this case, a malicious party would need the ability to scan some or all paper ballots prior to the election, to gain some knowledge of the order of these paper sheets, to observe the order that some or all voters receive paper ballots (and, if voters can randomly choose or shuffle the ballots, to draw meaningful inferences in spite of this uncertainty), and to fingerprint some or all of the resulting paper ballots later (or if the voting machine records the fingerprint-vote combinations electronically, to observe these values).

Some optical scan systems use paper ballots which contain a stub that includes a serial number. These stubs are removed for privacy reasons before a voter submits her ballot. Suppose that an adversary can scan ballots and associate fingerprints with each serial number prior to the election. If the adversary can observe the serial number of a ballot given to a voter, that adversary immediately knows the fingerprint of that voter's ballot. This attack emphasizes that, even if removed, serial numbers or other identifiable attributes on a ballot can threaten voter privacy[4].

Pre-Completed Ballots. Fingerprints can also enable dangerous coercion attacks utilizing pre-completed ballots. Imagine that an adversary wishes to coerce voters into choosing a particular candidate. The adversary can distribute precompleted ballots to targeted individuals, recording the fingerprint of the ballot provided to each voter. Alternatively, an adversary could distribute blank (fingerprinted) ballots to each targeted voter, along with instructions on how to vote. If the adversary has access to the ballots after the election, she can use the fingerprints to reidentify the

[3] Even if infeasible with present machines, it may become feasible as low-cost scanners increase in resolution and gain additional capabilities.

[4] Thank you to the anonymous reviewer for developing this interesting attack.

distributed ballots. This would allow the adversary to confirm that the provided ballots are in the ballot box and contain the “correct” votes.

Mitigation.

The various attack scenarios place certain requirements on an attacker. By making these requirements more difficult to achieve, the feasibility of these threats may be reduced.[5]

In all cases discussed in Section 4.1.1, an adversary must scan some or all paper ballots before voters cast those ballots. To prevent an adversary from producing scans, the paper sheets or rolls should be kept in a locked box whenever possible, and access to the paper should be monitored prior to election day. Election officials should make every effort to prevent the introduction of outside, rather than official, ballots into the ballot box. Scanners and computing devices should be kept away from the paper ballots.

Except in the case of DRE-VVPAT with paper spools, an adversary needs the ability to completely or partially learn the order of these paper sheets. This can be mitigated by shuffling the sheets immediately prior to election day. Depending on how well-shuffled the ballots are, the order information necessary to mount an attack may be destroyed. Shuffling is far from ideal, but it can make attacks harder.

These scenarios also rely on an adversary’s ability to observe the order that voters obtain their paper ballots or the order that these voters enter the voting booth. Given the need for poll workers and observers to view the process, we unfortunately cannot eliminate these possibilities. To mitigate the threats, however, voters should have the ability (if reasonably possible) to shuffle or otherwise re-randomize the pile of blank

[5] I do not consider mitigation techniques that seek to make paper ballots more difficult to fingerprint by modifying the ballots or using a different material. Even if possible, such solutions are likely both to be costly and to be vulnerable to future specialized attacks.

paper ballots. This protects other voters too, as randomization of ballots increases an adversary's uncertainly in the relationship between voters and fingerprints.

Alternatively, cryptographic techniques may assist in ensuring properly shuffled ballots. For example, see Xia et al. [106], who consider a similar problem. At this time, we have reservations regarding the practicality and security of applying such techniques to this problem, but existing or future techniques may prove themselves to be efficient and secure if carefully applied to this unique scenario.

In many cases, an adversary requires the ability to scan certain ballots following the election. Except as necessary for auditing and other processes, scanners should not be allowed near the ballots following the election. In general, used ballots should be stored securely in a monitored location.

As described earlier, some optical scan machines may record fingerprint-vote combinations electronically, and the adversary may use this information to reidentify voters' ballots. These values should be stored and revealed only to the degree necessary to conduct the election and audit processes. If the goal of recording ballot fingerprints is to facilitate efficient audits, it may be sufficient to record only the fingerprint, limiting the potential for revealing voter selections.

No countermeasure discussed in this section is perfect on its own. In addition, some countermeasures may inhibit other necessary goals, such as the ability to conduct a secure, efficient audit. Election officials may choose to focus their efforts on a limited number of feasible countermeasures to eliminate this threat. A cautious ballot shuffling process probably has the greatest potential to eliminate the threat of fingerprinting to ballot secrecy with minimal impact on the remainder of the election process.

4.1.2 Individual Ballot Fingerprinting

In this scenario, a voter reveals his votes to a third party, perhaps under coercion or to permit vote selling. Suppose that a coercer tells a voter to scan his paper ballot between receiving and submitting it and to return a fingerprint of the scan to the coercer. After the election, the coercer can verify fingerprints to identify the voter's ballot among the set of legitimate ballots. If the coercer does not find the ballot or if the ballot contains "incorrect" votes, the voter may face repercussions.

Similarly, a voter may sell her vote by providing a purchaser with a scan or fingerprint of her ballot. If a legitimate ballot exists with that fingerprint and contains the correct votes, the purchaser will pay the voter.

Mitigation

Attacks based on the ability of individual voters to fingerprint their ballots are difficult to prevent. If voters have the covert ability to measure inherent, unique physical properties of their own ballots, election officials are left with few methods of recourse. Our best option to mitigate this threat is disallowing scanners or similar devices near the ballots throughout the election process. While the prospect of a voter sneaking a scanner into the voting booth may seem far-fetched, handheld scanners, increasing-quality cell phone cameras, and other technological innovations are increasing the practicality of these attacks.

Arguably, voters have long had the ability to make their ballots stand out through unique write-in choices or combinations of votes. The threat of fingerprinting differs, however, because a fingerprinted ballot is reidentifiable even if no affirmative steps are taken to make the ballot unique.

In each of the voting systems described in this section, each suffers from the ability to identify blank ballots. This is not surprising, as this threat was not considered in the design of these systems. In practice, there does not appear to be a simple solution to prevent individuals from intentionally revealing their selections, e.g. by taking a picture of their completed ballot. The best practical solution is to limit access to cast ballots after the election, limiting the ability of adversaries to verify that their target voted “correctly”. On the other hand, severely limiting access to completed ballots may conflict with other goals, such as conducting a truly transparent election.

The correct balance between maintaining the secret ballot and election transparency is an important policy question. In determining the correct tradeoff, the impact of the auditing scheme used to verify election results must be considered. The next section describes the implications of paper fingerprinting on the ability to efficiently audit elections, and describes how existing systems can benefit from this new capability.

4.2 Auditing Techniques

The popularity of paper ballots is partially a consequence of concerns regarding flaws and vulnerabilities in computerized voting systems. Software cannot change a paper ballot already in a ballot box, making voter-verified paper ballots a popular mitigation strategy. These paper ballots are only useful if someone verifies that they are consistent with the electronically tabulated outcome.

Rather than recount all paper ballots to verify the election results, we can examine some subset of these ballots to draw statistical inferences about the election’s outcome. The most popular, widely used auditing approach is precinct-based auditing (e.g., [8–10, 85, 92]). With precinct-based auditing, officials and other parties randomly select some subset of election precincts. For the selected precincts, officials manually count the votes on all paper ballots and ensure that they match the electronic results.

Sampling at a finer level of granularity than precincts can allow for equally strong statistical inferences with fewer paper ballots manually reviewed. The finest level of granularity possible is ballot-based auditing, in which auditors select some subset of electronic ballots and ensure that they match their corresponding paper ballots (e.g., [21, 51, 73]). Ballot-based auditing presents a number of subtle challenges. For example, it can be difficult to ensure that ballots are selected at random without compromising ballot secrecy.

Calandrino et al. describe a machine-assisted auditing method that allows ballot-based auditing yet strives to preserve ballot secrecy [21]. Following the election, an auditing machine prints serial numbers on paper ballots and rescans the ballots, storing the serial numbers and votes electronically. If these votes sum to the initially reported electronic tally, auditors randomly select electronic ballots based on serial number and manually verify that they match the corresponding paper ballots. The scheme in [21] is able to detect discrepancies even if the auditing machine misbehaves. We propose a method that allows ballot-based auditing without printing serial numbers on ballots, instead relying on paper ballot fingerprints.

For our auditing method, we make several assumptions. First, we assume that precincts maintain a sign-in list that observers may monitor as voters enter and leave a polling place. As is standard practice, the sign-in list is made public after the election. This list allows anyone to determine an accurate count of the number of voters. We assume that every voter signing in casts a ballot. In practice, voters rarely sign in without casting a ballot, and we leave methods for ascertaining the number of ballots cast to future work (any issues with obtaining an accurate count affects all known auditing schemes, not just the ones in this chapter). Our auditing scheme checks for discrepancies not only between the paper and electronic records but between those records and the totals from the sign-in list.

Many ballot-based auditing schemes assume that the set of paper ballots contains no added ballots. Section 4.2.2 describes serious additional threats that are possible if an adversary is able to add paper ballots to the ballot box—even if the set of correct paper ballots remains. These attacks can be difficult to detect, traditionally requiring an accurate count of the paper ballots. We discuss this threat and describe how to use fingerprints to detect added paper ballots more efficiently.

Auditors might also want the ability to verify that the set of paper ballots in the ballot box matches the ones delivered to the polling place before the election. This capability enables a number of possibilities. For example, officials could find the set of legitimate paper ballots in the event of ballot-box stuffing. Section 4.2.3 describes how to perform this check by fingerprinting ballots immediately prior to the election.

4.2.1 Fingerprint-Based Auditing

We first discuss a scheme for using fingerprints to detect mismatches from the electronic ballots to the corresponding paper ballots in the ballot box. Our process is designed such that, if at least B incorrect electronic ballots exist, we will find one or more discrepancies with probability greater than or equal to a desired confidence level c. This scheme is primarily for demonstrative purposes, as it requires revelation of the fingerprint and full combination of votes for each ballot cast, potentially posing serious privacy concerns. [6] This requirement undermines ballot secrecy if a voter can distinguish her ballot through her votes.

Throughout this section, we assume the use of an optical scan voting machine. The scheme works as follows. When a voter submits her paper ballot, the voting machine records both the ballot's fingerprint and a vector of the votes contained on that ballot. At the end of the election, officials immediately publish both these fingerprint-vote pairs and the voter sign-in sheet (as discussed earlier, the sign-in sheet should be public throughout election day). The published list of fingerprint-vote pairs must be in history-independent order, e.g. sorted lexicographically by fingerprint

[6] In practice, we want the audit process to determine whether we can be confident in the election's outcome even if we observe a small number of discrepancies. Although we use the more conventional goal of election auditing in this chapter, these methods can extend to meet the more ambitious practical goal.

value, in order to minimize information revealed to an adversary about voter order. Ideally, the list should be maintained in a history-hiding data structure [13]. While the published list of fingerprints may be independent of voter order, our biometric may still allow some bits to be revealed, e.g. through steganographic means [33]. Revealing fingerprint-vote pairs allows any citizen to confirm that the number of electronic ballots matches the number of signed-in voters and that the votes posted add up to the reported tallies. Our final step is to sample from the electronic ballots and ensure that matching paper ballots exist. Otherwise, a discrepancy exists. Note that switching a vote on a single ballot from Candidate A to Candidate B affects the margin between the candidates by two: Candidate A loses one vote and Candidate B gains one vote. Therefore, we seek to reject the hypothesis that $B = \lceil margin/2 \rceil$ incorrect electronic ballots exist.

To sample from the electronic ballots, we make a list of the (precinct, fingerprint, vote vector) triples, one for each ballot, ordered lexicographically. We will sample items from this list and ensure that a ballot containing the proper fingerprint and vote vector exists in the given precinct for each item selected. Two possible sampling methods exist, which we adapt from [9, 10, 21]. The first option is to sample a fixed number of items from this list. We call this the fixed sample size method. Given N total reported ballots, a minimum of $B = \lceil margin/2 \rceil$ incorrect electronic ballots, and a desired confidence level of c, we require a minimum sample size, n, of:

$$n = \min \left\{ u \mid 1 - \prod_{k=0}^{u-1} \frac{N - B - k}{N - k} \geq c \right\}$$

Alternatively, we may select each electronic ballot independently with probability p, where $p \geq 1 - (1 - c)^{1/B}$. This latter approach yields a variable sample size and results in marginally more ballots selected on average, but it is more amenable to optimizations, as discussed below. We call this the variable sample size method.

General Election					
Issue	Totals		Fixed SS*	Varying SS*	Pct-Based
	# Votes	Margin	# Bal (Man)	# Bal (Man)	# Bal (Man)
U.S. Senate	2,370,445	0.39%	2,337	2,339	1,141,900
Const. Amnd.	2,328,224	14.12%	63	65	8,062
U.S. House	173,159	2.82%	325	327	62,469
U.S. House	212,079	19.19%	46	48	1,958
U.S. House	241,134	16.36%	54	56	6,120
U.S. House	235,280	11.88%	76	77	12,991
Delegate	14,963	5.75%	157	159	11,442
Average	796,469	8.11%	437	439	177,849

Table 4.1: Ballot-Based Auditing with Fingerprints (For Both Fixed and Varying Sample Size Methods) vs. Precinct-Based Auditing on 2006 Virginia General Election Data. (Note: these numbers are originally from [21], which describes post-election auditing methods that yield equivalent sample sizes to the methods in this chapter.) SS* = Sample Size

Given n or p, a number of existing papers describe how to securely and efficiently sample from a list of items, and we refer the reader to those papers for greater detail (see [20, 21, 28]).

When a ballot is sampled for audit, auditors feed the ballots from that precinct into a scanner. The scanner stops when it observes a match for that ballot's electronically reported fingerprint. All auditors and observers may verify that the vote vectors match, and observers may use their own scanners to verify the fingerprint. Because this check only verifies that a paper ballot exists matching each sampled electronic ballot, observers in this process only need the ability to personally scan the paper ballots that reportedly match the selected electronic ones—not all paper ballots—making the process far more efficient.

Table 4.1 compares the number of ballots manually reviewed with these methods to the number manually reviewed with precinct-based auditing (using the methods in [92]) for races with margin under 20% in the 2006 Virginia general election. For example, in Virginia's 2006 Webb-Allen U.S. Senate race with a margin of 0.39%, the fixed sample size fingerprinting method requires manual review of 2,337 of 2,370,445 ballots, and the varying sample size method requires review of 2,339 ballots (on average). Precinct-based auditing requires manual review of 1,141,900 ballots (on average).

Additional techniques are possible to reduce the number of ballots to be sampled, potentially resulting in dramatic efficiency gains. For example, we may take the reported contents of ballots into account when determining the probability that we review each of those ballots. Given that all vote vectors must be public for this auditing method, techniques that consider ballot contents are straightforward to apply. See [21] for details.

This auditing process requires that fingerprint-vote vector combinations be released and that scanners be allowed near the ballots following the election. As discussed earlier in this section and in Section 4.1, these choices may enable certain attack scenarios, so officials must carefully utilize other countermeasures to ensure ballot secrecy. It is important to note that only *audited* ballots need to have their fingerprints published. In addition, the practical efficiency and simplicity of this process are unclear and require additional testing. As an alternative, one could check the electronic ballot to paper ballot correspondence using machine-assisted auditing [21], reducing the number of ballots for which the vote combination is revealed publicly (with that technique, only precincts containing sampled ballots must reveal vote vectors). The techniques in the following sections would remain applicable.

4.2.2 A Paper-to-Electronic Check

Suppose that, in addition to an ability to change electronic records, an adversary has the ability to add paper ballots to the ballot box. This may be true for a number of reasons. For example, a compromised DRE may print additional paper ballots or an official may push extra ballots through the slot of a locked ballot box. The previous check verifies that each electronic ballot has a corresponding paper ballot. If an adversary can add paper ballots, however, the "corresponding paper ballots" might be fraudulently added ballots while the legitimate ballots may be excluded from the electronic results. This would allow an adversary to steal all electronic ballots in a precinct, yet every electronic ballot would have a matching paper ballot (the ballot box would just have many extra unmatching ballots). In this case, we must check not only that each electronic ballot has a corresponding paper ballot but also that each paper ballot has a corresponding electronic ballot. Traditionally, this would require a count of the number of paper ballots in all precincts, but fingerprinting can allow more efficient approaches.

The most practical approach to detect fraud is to force an adversary to only perform electronic fraud by making paper-based fraud too risky. An adversary who only commits electronic fraud provides a number of benefits. First, it forces the adversary to cheat in more places. It is straight forward to ensure the number of electronic votes does not exceed the number of registered voters who signed-in at a particular precinct. Auditing can also ensure that each ballot in the ballot box has a corresponding electronic record. This provides a mechanism for detecting ballot box stuffing. Finally, it forces the adversary to commit to the fraud prior to an audit beginning.

Note that we ignore an adversary who also has the ability to remove paper ballots from the ballot box. Election procedures often dictate that the ballot box should only be unlocked under the watchful eyes of observers. Further, an adversary that can also remove ballots could commit fraud that would be undetectable from the paper and electronic records alone. Such an adversary could arbitrarily modify the paper ballots to match any fraudulent electronic results. While this section focuses on adversaries who can commit electronic fraud, the next section focuses on how to detect an adversary who can only attack paper ballots, and how this attack can be detected.

As in the previous section, we assume use of an optical scan voting machine. When a voter submits her paper ballot, the voting machine records the ballot's fingerprint (but not the vote-vector, so voter privacy is protected). At the end of election day, officials publish both the fingerprints and the voter sign-in sheet. Any interested party can confirm that the reported number of ballots matches the number of voters. As the final step, election officials and other interested parties will have the opportunity to select paper ballots from certain precincts' ballot boxes and ensure that they match the published fingerprints from those precincts. The key property is that we can achieve the desired level of confidence in the election's outcome as long as any participant selects randomly from the ballot box.

When participants perform this sampling, if they see a ballot not on the reported list of fingerprints for the precinct, this indicates that additional ballots are in the ballot box. The security of this scheme rests on an adversary's inability to produce another ballot with the same fingerprint as a legitimate ballot. These methods rely on difficult-to-duplicate properties of paper and are not possible with printed serial numbers. If we sample a ballot with serial number 10 from a ballot box, no guarantee exists that another ballot with the same serial number is not in the same box. If we sample a ballot with a given fingerprint from a ballot box, we may reasonably believe that no additional ballots with the same fingerprint are in the same box, even if an adversary has attempted to undermine this property.

4.2.3 Pre-Scanning Ballots

An additional property that elections officials may wish to check is whether the set of paper ballots in the ballot box is the same as, or a subset of, the set of ballots delivered to the precinct on election day. To do so, officials may pre-scan and publish fingerprints for the paper ballots prior to election day. Immediately prior to the election, election officials, candidates, etc. may verify that these published fingerprints are correct by checking that paper ballots exist matching the published fingerprints and that no extra paper ballots exist (using methods described in the previous two sections). In this way, participants may draw statistically strong conclusions that the set of paper ballots matches the published fingerprints.

When voters submit their paper ballots, the optical scan machine may store fingerprints for the ballots.[7] Following the election, officials may sample fingerprints and paper ballots to verify that legitimate ballots ended up in the ballot box and that the set of reportedly unused ballots were actually unused. Among other things, this check would allow auditors to isolate the set of legitimate ballots in a ballot box, helping to deter attacks that rely on slipping in fake ballots.

4.3 Analysis of an End-to-End Scheme

While the primary focus is on widely deployed voting systems, it is important to briefly discuss the impact of paper fingerprinting on an existing end-to-end voting scheme. A full analysis of these consequences (particularly positive consequences) is somewhat system-specific. The analysis suggests, however, that end-to-end systems must demonstrate caution if relying on paper.

Scantegrity II. Chaum et al. present a system providing end-to-end verifiability: Scantegrity II [25]. It is an interesting proposal, and it provides for a useful case study.

The basic Scantegrity II system relies on a process similar to traditional optical scan voting, but selection of a candidate causes certain codes written in invisible ink

[7] The machine could even reject ballots with invalid fingerprints, though officials should not trust that the machine performs this check.

to appear on the paper ballot. Voters may copy the codes for chosen candidates to a paper tab that is to be removed from the ballot and serves as a receipt. These paper tabs contain serial numbers that match serial numbers printed on the paper ballots themselves. When voters cast their ballots, the relationships from voters to codes to candidates is partially destroyed, and the ballots are kept in a locked box (see [25] for additional details and explanation). The system relies on these security measures to prevent an adversary from recovering a voter's ballot or otherwise learning the choices on that ballot.

By keeping paper ballots secure at all times following submission of the ballots, the Scantegrity II system provides a certain level of defense against attacks utilizing paper fingerprinting. If paper ballots are not revealed after voting concludes, an adversary cannot reassociate those ballots with pre-computed fingerprints. Although some risk exists if this assumption of physical security is violated, the serial numbers printed on ballots pose a far greater risk than fingerprints. Overall, the assumptions of the basic Scantegrity II system increase the difficulty of fingerprinting-based attacks, but a violation of these assumptions would raise the possibility of attacks like those against any other optical scan system.

Paper fingerprinting can present a significant benefit in this context, however. One threat to this system is if an adversary can introduce false ballots with the same code corresponding to multiple candidates (see [25]). Using the techniques in Section 4.2.3, officials may assemble a list of valid ballot fingerprints prior to election day. In the event that a voter receives a fraudulent ballot, the optical scan machine could detect and reject the unexpected ballot.

While fingerprinting can pose serious risks to the Scantegrity II system if assumptions are violated, it also presents an opportunity to improve on existing work. The benefits and drawbacks of fingerprinting will vary dramatically based on details of the end-to-end system. As a general rule, however, any end-to-end system that relies on the uniformity of paper to provide security is at risk. Nevertheless, careful analysis may reveal mitigation strategies and even ways in which paper fingerprinting can strengthen these systems.

4.4 Discussion

Paper fingerprinting poses both challenges and opportunities for election officials. This chapter outlined several threats to ballot secrecy due to paper identification and suggested some mitigation strategies to counter these threats. While the most obvious consequences of paper identification are negative, it can also help improve election integrity. Fingerprints can enable an efficient post-election audit process and help detect and prevent additional threats to election integrity.

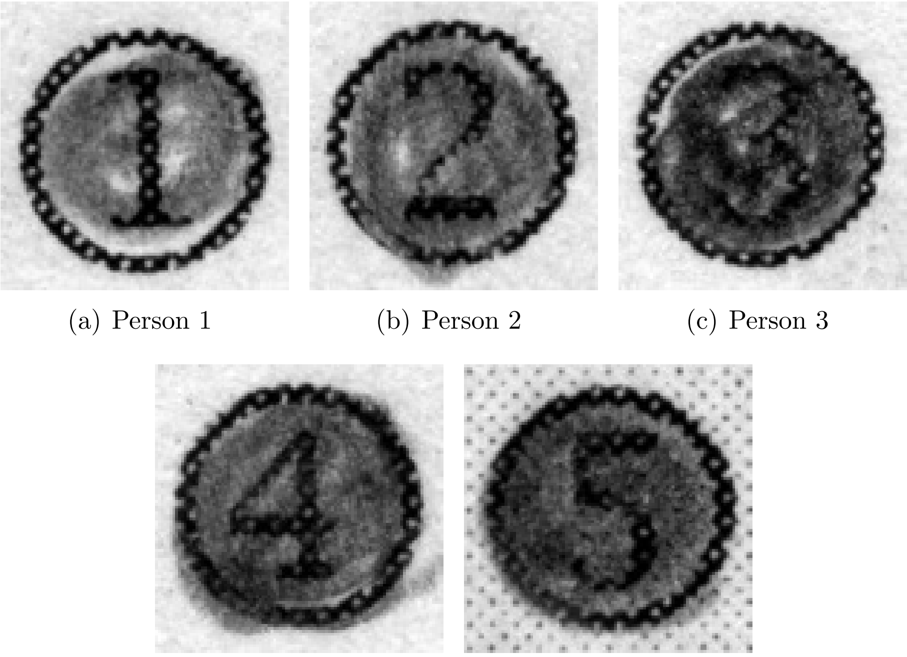
As technology and algorithms improve, it may be possible for digital cameras and other handheld devices to fingerprint ballots. These new advances will pose additional risks to ballot privacy and should be addressed by future work. In the near future, however, paper will likely remain a critical component of the voting process due to its reliability, cost, familiarity to the public, and ability to stymie many threats to electronic voting systems.

Chapter 5

Bubble Trouble: Off-Line De-Anonymization of Bubble Forms

Scantron-style fill-in-the-bubble forms are a popular means of obtaining human responses to multiple-choice questions. Whether conducting surveys, academic tests, or elections, these forms allow straightforward user completion and fast, accurate machine input. Although not every use of bubble forms demands anonymity, common perception suggests that bubble completion does not result in distinctive marks. This chapter challenges this assumption, demonstrating that it is false under certain scenarios, enabling use of these markings as a biometric. The ability to uncover identifying bubble marking patterns has far-reaching potential implications, from detecting cheating on standardized tests to threatening the anonymity of election ballots.

Bubble forms are widely used in scenarios where confirming or protecting the identity of respondents is critical. Over 137 million registered voters in the United States reside in precincts with optical scan voting machines [101], which traditionally use fill-in-the-bubble paper ballots. Voter privacy (and certain forms of fraud) relies on an inability to connect voters with these ballots. Surveys for research and other purposes use bubble forms to automate data collection. The anonymity of survey subjects not only affects subject honesty but also impacts requirements governing human subjects research [99]. Over 1.6 million members of the high school class of 2010 completed the SAT [27], one of many large-scale standardized tests using bubble sheets. Educators, testing services, and other stakeholders have incentives to detect cheating on these tests. The implications of these findings extend to any use of bubble forms for which the ability to "fingerprint" respondents may have consequences, positive or negative.



(a) Person 1 (b) Person 2 (c) Person 3

(d) Person 4 (e) Person 4 - Gray

Figure 5.1: Example marked bubbles. The background color is white in all examples except Figure 5.1(e), which is gray.

The General Framework provides a high-level outline for examining the assumption that marked bubbles do not convey information that can identify the respondent, see Figure 5.2. To evaluate this assumption, we use a desktop scanner to capture digital images of documents containing marked bubbles and extract the portions which

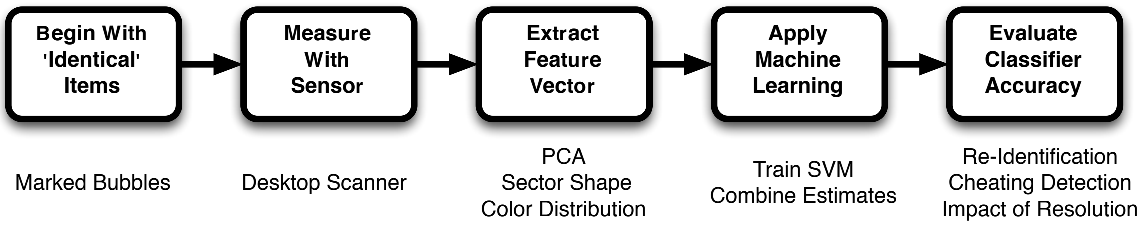


Figure 5.2: Application of the General Framework, introduced in Chapter 2, to identifying the respondent responsible for marking a bubble.

contain marked bubbles (Stage 2 of the General Framework). In Section 5.1.1 (Stage 3), we extract a set of features that capture the unique properties of a marked bubble. We then apply standard machine learning techniques in Section 5.1.2 (Stage 4) to distinguish between respondents based on the properties of their marked bubbles.

To evaluate our results on real-world data, we use a corpus of over ninety answer sheets from a survey of high school students in Section 5.2 (Stage 5). For certain parameters, our algorithms' top match is correct over 50% percent of the time, and the correct respondent falls in the top 3 matches 75% percent of the time. In addition, we test our ability to detect when someone other than the expected respondent completes a form, simultaneously achieving false positive and false negative rates below 10%. We conduct limited additional tests to confirm our results and explore details available from bubble markings.

Depending on the application, these techniques can have positive or negative repercussions (see Section 5.3). Analysis of answer sheets for standardized tests could provide evidence of cheating by test-takers, proctors, or other parties. Similarly, scrutiny of optical-scan ballots could uncover evidence of ballot-box stuffing and other forms of election fraud. With further improvements in accuracy, the methods introduced here could even enable new forms of authentication. Unfortunately, the techniques could also undermine the secret ballot and anonymous surveys. For example, some jurisdictions publish scanned images of ballots following elections, and employers could match these publicly available ballots against bubble-form employment applications. Bubble markings serve as a biometric even on forms and surveys otherwise containing no explicitly identifying information. We discuss methods for minimizing the negative impact of this work while exploiting its positive uses (see Section 5.4).

Because our test data is somewhat limited, we discuss the value of future additional tests (see Section 5.6). For example, longitudinal data would allow us to better understand the stability of an individual's distinguishing features over time, and stability is critical for most uses discussed in the previous paragraph.

5.1 Learning Distinctive Features

Filling in a bubble is a narrow, straightforward task. Consequently, the space for inadvertent variation is relatively constrained. The major characteristics of a filled-in bubble are consistent across the image population—most are relatively circular and dark in similar locations with slight imperfections—resulting in a largely homogeneous set. See Figure 5.1. This creates a challenge in capturing the unique qualities of each bubble and extrapolating a respondent's identity from them.

We assume that all respondents start from the same original state—an empty bubble with a number inscribed corresponding to the answer choice (e.g., choices 1-5 in Figure 5.1). When respondents fill in a bubble, opportunities for variation include the pressure applied to the drawing instrument, the drawing motions employed, and the care demonstrated in uniformly darkening the entire bubble. In certain contexts, such as signature verification, these details can be useful. In this work, we consider applications for which it would be infeasible to monitor the exact position, pressure, and velocity of pencil motions throughout the coloring process. These variations in motion and pressure manifest as visible differences in marked bubbles. When examining the resulting static images, our focus is to identify the characteristic features of markings made by each respondent. Should access to dynamic information be possible, it would only strengthen our results. We consider the implications of dynamic information for bubble-based authentication in Section 5.3.

5.1.1 Generating a Bubble Feature Vector

Image recognition techniques often use feature vectors to concisely represent the important characteristics of an image. As applied to bubbles, a feature vector should capture the unique ways that a mark differs from a perfectly completed bubble, focusing on characteristics that tend to distinguish respondents. Because completed bubbles tend to be relatively homogeneous in shape, many common metrics do not work well here. To measure the unique qualities, we generate a feature vector that blends several approaches from the image recognition literature. Specifically, we use PCA, shape descriptors, and a custom bubble color distribution to generate a feature vector for each image.

Principal Component Analysis (PCA) is one common technique for generating a feature set to represent an image [52]. At a high level, PCA reduces the dimensionality of an image, generating a concise set of features that are statistically independent from one another. PCA begins with a sample set of representative images to generate a set of eigenvectors. In most of our experiments, the representative set was comprised of 368 images and contained at least one image for each (respondent, answer choice) pair. Each representative image is normalized and treated as a column in a matrix. PCA extracts a set of eigenvectors from this matrix, forming a basis. We retain the 100 eigenvectors with the highest weight. These eigenvectors account for at least 90% of the information contained in the representative images.

To generate the PCA segment of our feature vector, a normalized input image (treated as a column vector) is projected onto the basis defined by the 100 strongest eigenvectors. The feature vector is the image's coordinates in this vector space—i.e.,

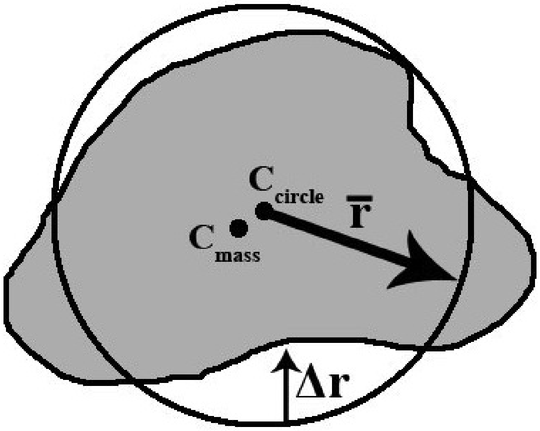


Figure 5.3: An example bubble marking with an approximating circle. The circle minimizes the sum of the squared deviation from the radius. We calculate the circle's center and mean radius, the marking's variance from the radius, and the marking's center of mass.

the weights on the eigenvectors. Because PCA is such a general technique, it may fail to capture certain context-specific geometric characteristics when working exclusively with marked bubbles.

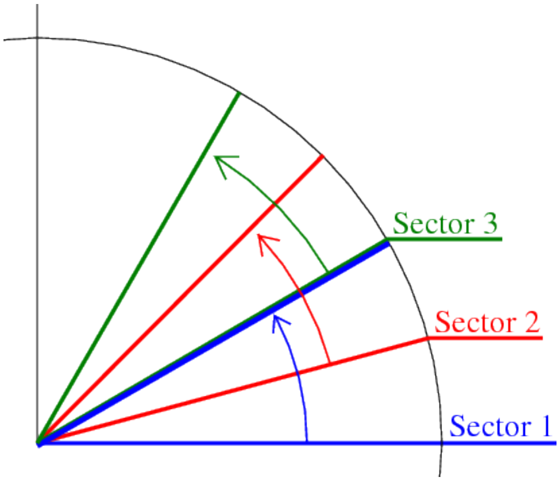


Figure 5.4: Each dot is split into twenty-four 15° slices. Adjacent slices are combined to form a sector, spanning 30°. The first few sectors are depicted here.

To compensate for the limitations of PCA, we capture shape details of each bubble using a set of geometric descriptors and capture color variations using a custom metric. Peura et al. [81] describe a diverse a set of geometric descriptors that measure

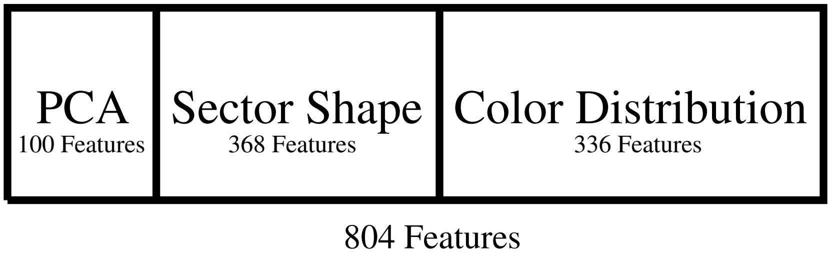


Figure 5.5: Feature vector components and their contributions to the final feature vector length.

statistics about various shapes. This set includes a shape's center of mass, the center and radius of a circle approximating its shape, and variance of the shape from the approximating circle's radius (see Figure 5.3). The approximating circle minimizes the sum of squared radius deviations. We apply the specified descriptors to capture properties of a marked bubble's boundary. Instead of generating these descriptors for the full marked bubble alone, we also generate the center of mass, mean radius, and radial variance for "sectors" of the marked bubble. To form these sectors, we first evenly divide each dot into twenty-four 15° "slices." Sectors are the 24 overlapping pairs of adjacent slices (see Figure 5.4). Together, these geometric descriptors add 368 features.

Finally, we developed and use a simple custom metric to represent color details. We divide a dot into sectors as in the previous paragraph. For each sector, we create a histogram of the grayscale values for the sector consisting of fifteen buckets. We throw away the darkest bucket, as these pixels often represent the black ink of the circle border and answer choice numbering. Color distribution therefore adds an additional 14 features for each sector, or a total of 336 additional features.

The resulting feature vector consists of 804 features that describe shape and color details for a dot and each of its constituent sectors (see Figure 5.5). See Section 5.2.3, where we evaluate the benefits of this combination of features. Given feature vectors, we can apply machine learning techniques to infer distinguishing details and differentiate between individuals.

5.1.2 Identifying Distinguishing Features

Once a set of feature vectors are generated for the relevant dots, we use machine learning to identify and utilize the important features. Our analysis tools make heavy use of Weka, a popular Java-based machine learning workbench that provides a variety of pre-implemented learning methods [46]. In all experiments, we used Weka version 3.6.3.

We apply Weka's implementation of the Sequential Minimal Optimization (SMO) supervised learning algorithm to infer distinctive features of respondents and classify images. SMO is an efficient method for training support vector machines [82]. Weka can accept a training dataset as input, use the training set and learning algorithm to create a model, and evaluate the model on a test set. In classifying individual data points, Weka internally generates a distribution over possible classes, choosing the class with the highest weight. For us, this distribution is useful in ranking the respondents believed to be responsible for a dot. We built glue code to collect and process both internal and exposed Weka data efficiently.

5.2 Evaluation

To evaluate our methods, we obtained a corpus of 154 surveys distributed to high school students for research unrelated to our study. Although each survey is ten pages, the first page contained direct identifying information and was removed prior to our access. Each of the nine available pages contains approximately ten questions, and each question has five possible answers, selected by completing round bubbles numbered 1-5 (as shown in Figure 5.1).

From the corpus of surveys, we removed any completed in pen to avoid training on writing utensil or pen color.[1] Because answer choices are numbered, some risk exists of training on answer choice rather than marking patterns—e.g., respondent X tends to select bubbles with "4" in the background. For readability, survey questions alternate between a white background and a gray background. To avoid training bias, we included only surveys containing at least five choices for each answer 1-4 on a white background (except where stated otherwise), leaving us with 92 surveys.

For the 92 surveys meeting our criteria, we scanned the documents using an Epson v700 Scanner at 1200 DPI. We developed tools to automatically identify, extract, and label marked bubbles by question answered and choice selected. After running these tools on the scanned images, we manually inspected the resulting images to ensure accurate extraction and labeling.

Due to the criteria that we imposed on the surveys, each survey considered has at least twenty marked bubbles on a white background, with five bubbles for the "1" answer, five for the "2" answer, five for the "3" answer, and five for the "4" answer.[2] For each experiment, we selected our training and test sets randomly from this set of twenty bubbles, ensuring that sets have equal numbers of "1," "2," "3," and "4" bubbles for each respondent and trying to balance the number of bubbles for each answer choice when possible.

In all experiments, a random subset of the training set was selected and used to generate eigenvectors for PCA. We required that this subset contain at least one example from each respondent for each of the four relevant answer choices but placed no additional constraints on selection. For each dot in the training and test sets, we

[1] We note that respondents failing to use pencil or to complete the survey anecdotally tended not to be cautious about filling in the bubbles completely. Therefore, these respondents may be more distinguishable than those whose surveys were included in our experiment.

[2] To keep a relatively large number of surveys, we did not consider the number of "5" answers and do not use these answers in our analysis.

generated a feature vector using PCA, geometric descriptors, and color distribution, as described in Section 5.1.1.

We conducted two primary experiments and a number of complementary experiments. The first major test explores our ability to re-identify a respondent from a test set of eight marks given a training set of twelve marks per respondent. The second evaluates our ability to detect when someone other than the official respondent completes a bubble form. To investigate the potential of bubble markings and confirm our results, we conducted seven additional experiments. We repeated each experiment ten times and report the average of these runs.

Recall from Section 5.1.2 that we can rank the respondents based on how likely we believe they are to be responsible for marking a particular bubble. For example, the respondent that created a dot could be the first choice or fiftieth choice of our algorithms. A number of our graphs effectively plot a cumulative distribution showing the percent of test cases for which the true corresponding respondent falls at or above a certain rank—e.g., for 75% of respondents in the test set, the respondent's true identity is in the top three guesses.

5.2.1 Respondent Re-Identification

This experiment measured the ability to re-identify individuals from their bubble marking patterns. For this test, we trained our model using twelve sample bubbles per respondent, including three bubbles for each answer choice 1-4. Our test set for each respondent contained the remaining two bubbles for each answer choice, for a total of eight test bubbles per respondent. We applied the trained model to each of the 92 respondents' test sets and determined whether the predicted identity was correct.

To use multiple marks per respondent in the test set, we classify the marks individually, yielding a distribution over the respondents for each mark in the set. After

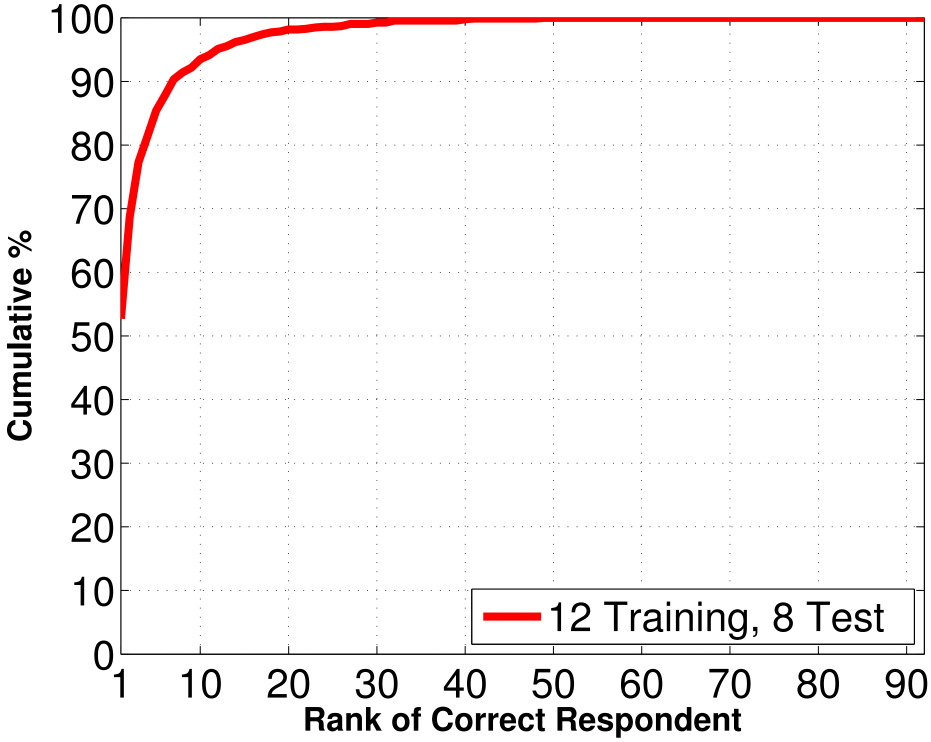


Figure 5.6: Respondent re-identification with 12 training bubbles and 8 test bubbles per respondent.

obtaining the distribution for each test bubble in a group, we combine this data by averaging the likelihoods that the bubbles correspond to each respondent.[3] Our algorithms then order the respondents from highest to lowest average confidence, with highest confidence corresponding to the top choice.

On average, our algorithm's first guess identified the correct respondent with 51.1% accuracy. The correct respondent fell in the top three guesses 75.0% of the time and in the top ten guesses 92.4% of the time. See Figure 5.6, which shows the percentage of test bubbles for which the correct respondent fell at or above each possible rank. This initial result suggests that individuals complete bubbles in a highly distinguishing manner, allowing re-identification with surprisingly high accuracy.

[3] Alternative methods of combining likelihood estimates, such as multiplying or adding, were considered and evaluated. However, averaging provided the most accurate predictions. This is likely due to the attenuated effect of outliers in the set of likelihoods.

Evaluating Single Contest Outcomes

In some circumstances it is desirable to estimate the likelihood that a respondent voted a particular way in a single contest. For example, suppose Bob gives his employees an 'anonymous' survey consisting of a number of true/false questions. At the completion of the survey Bob wishes to know the likelihood that one of his employees, Alice, voted a particular way on a single question. Bob doesn't know exactly which survey was completed by Alice – after all the survey was designed to be anonymous. To calculate the likelihood that Alice voted in a particular way on a particular question, Bob weighs the selection made on each survey against the likelihood that the corresponding survey belongs to Alice according to Equation 5.1.

$$\begin{aligned} S &= \{s|s_i = True\} \\ P(Alice\ Voted\ True) &= \sum_{s \epsilon S} P(s = Alice) \end{aligned} \tag{5.1}$$

The outcome of this method generates results consistent with what one expects in the absence of information about the respondents identity. For example, if Bob has no knowledge about which survey is Alice's, e.g. he has a uniform distribution of confidences across the surveys, then the the result is simply the percentage of surveys which voted a certain way in the selected contest.

5.2.2 Detecting Unauthorized Respondents

One possible application of this technique is to detect when someone other than the authorized respondent creates a set of bubbles. For example, another person might take a test or survey in place of an authorized respondent. We examined our ability to detect these cases by measuring how often our algorithm would correctly detect a fraudulent respondent who has claimed to be another respondent. We trained our model using twelve training samples from each respondent and examined the output of our model when presented with eight test bubbles. The distribution of these sets is the same as in Section 5.2.1.

For these tests, we set a threshold for the lowest rank accepted as the respondent. For example, suppose that the threshold is 12. To determine whether a given set of test bubbles would be accepted for a given respondent, we apply our trained model to the test set. If the respondent's identity appears in any of the top 12 (of 92) positions in the ranked list of respondents, that test set would be accepted for the respondent. For each respondent, we apply the trained model both to the respondent's own test bubbles and to the 91 other respondents' test bubbles.

We used two metrics to assess the performance of our algorithms in this scenario. The first, false positive rate, measures the probability that a given respondent would be rejected (labeled a cheater) for bubbles that the respondent actually completed. The second metric, false negative rate, measures the probability that bubbles completed by any of the 91 other respondents would be accepted as the true respondent's. We varied the threshold from 1 to 92 for our tests. We expected the relationship between threshold and false negative rate to be roughly linear: as the threshold increases by 1 this adds an additional candidate, which will almost always be false, increasing the false negative rate by $\frac{1}{92}$.[4]

Our results are presented in Figure 5.7. As we increase the threshold, the false positive rate drops precipitously while the false negative rate increases roughly linearly. If we increase the threshold to 8, then a fraudulent respondent has a 7.8% chance of avoiding detection (by being classified as the true respondent), while the true respondent has a 9.9% chance of being mislabeled a cheater. These error rates intersect with a threshold approximately equal to 9, where the false positive and false negative rates are 8.8%.

[4] This is not exact because the order of these rankings is not entirely random. After all, we seek to rank a respondent as highly as possible for the respondent's own test set.

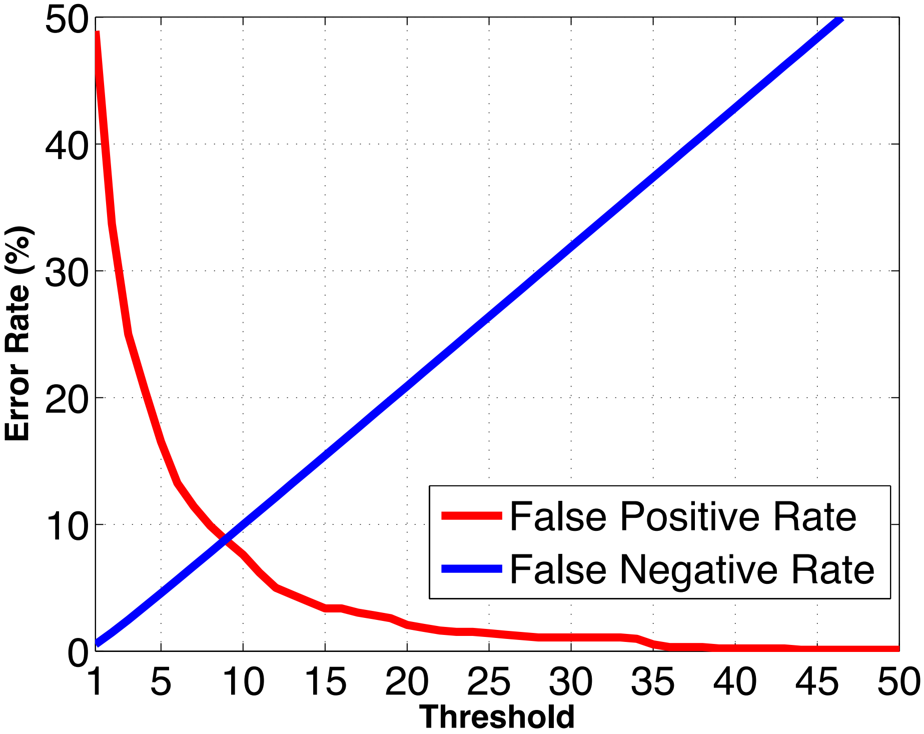


Figure 5.7: False positive and false negative rates when detecting unauthorized respondents.

5.2.3 Additional Experiments

To study the information conveyed by bubble markings and support our results, we performed seven complementary experiments. In the first, we evaluate the effect that scanner resolution has on re-identification accuracy. Next, we considered our ability to re-identify a respondent from a single test mark given a training set containing a single training mark from each respondent. Because bubble forms typically contain multiple markings, this experiment is somewhat artificial, but it hints at the information available from a single dot. The third and fourth supplemental experiments explored the benefits of increasing the training and test set sizes respectively while holding the other set to a single bubble. In the fifth test, we examined the tradeoff between training and test set sizes. The final two experiments validated our results

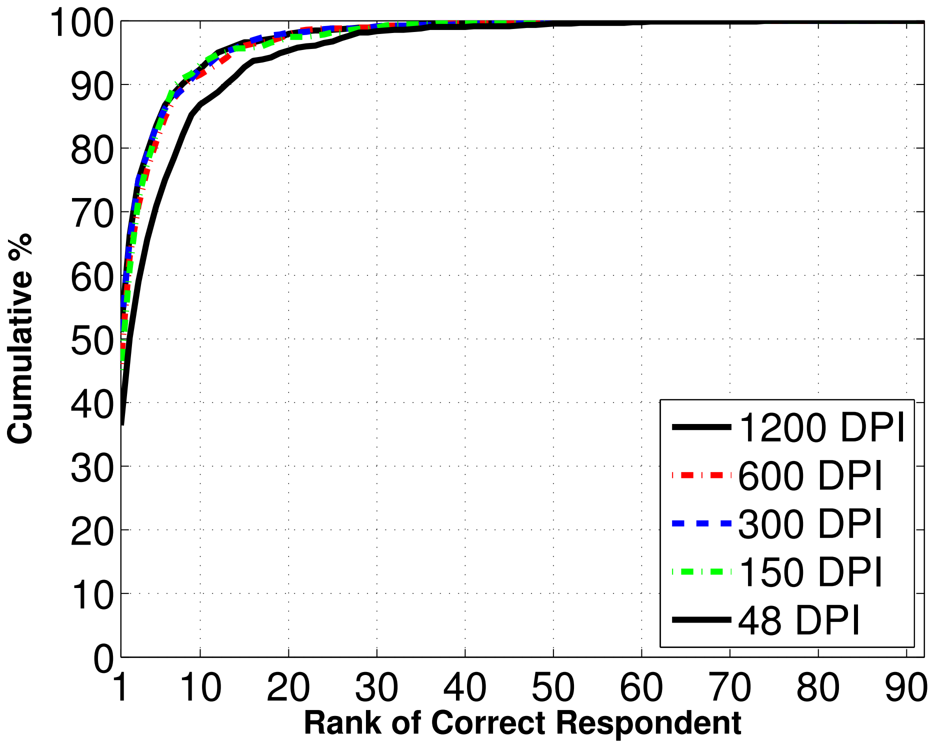


Figure 5.8: Respondent re-identification accuracy using lower-resolution images. Note that the 1200, 600, 300, and 150 DPI lines almost entirely overlap.

using additional gray bubbles from the sample surveys and demonstrated the benefits of our hybrid feature set compared to PCA alone. As with the primary experiments, we repeated each experiment ten times.

Effect of resolution on accuracy. In practice, high-resolution scans of bubble forms may not be available, but access to lower resolution scans may be feasible. To determine the impact of resolution on re-identification accuracy, we down-sampled each ballot from the original 1200 DPI to 600, 300, 150, and 48 DPI. We then repeated the re-identification experiment of Section 5.2.1 on bubbles at each resolution.

Figure 5.8 shows that decreasing the image resolution has little impact on performance for resolutions above 150 DPI. At 150 DPI, the accuracy of our algorithm's first guess decreases to 45.1% from the 51.1% accuracy observed at 1200 DPI. Accuracy remains relatively strong even at 48 DPI, with the first guess correct 36.4%

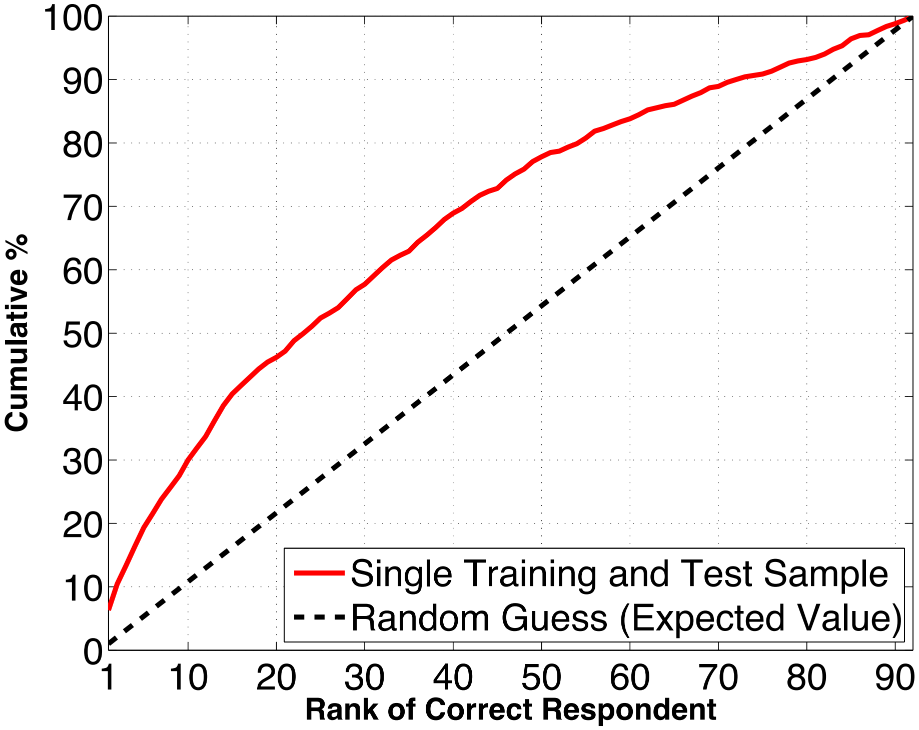


Figure 5.9: One marked bubble per respondent in each of the training and test sets. The expected value from random guessing is provided as reference.

of the time and the correct respondent falling in the top ten guesses 86.8% of the time. While down-sampling may not perfectly replicate scanning at a lower resolution, these results suggest that strong accuracy remains feasible even at resolutions for which printed text is difficult to read.

Single bubble re-identification. This experiment measured the ability to re-identify an individual using a single marked bubble in the test set and a single example per respondent in the training set. This is a worst-case scenario, as bubble forms typically contain multiple markings. We extracted two bubbles from each survey and trained a model using the first bubble.[5] We then applied the trained model to each of the 92 second bubbles and determined whether the predicted identity was cor-

[5] Note: In this experiment, we removed the restriction that the set of images used to generate eigenvectors for PCA contains an example from each column.

rect. Under these constrained circumstances, an accuracy rate above that of random guessing ($\frac{1}{92} \approx 1.087\%$) would suggest that marked bubbles embed distinguishing features.

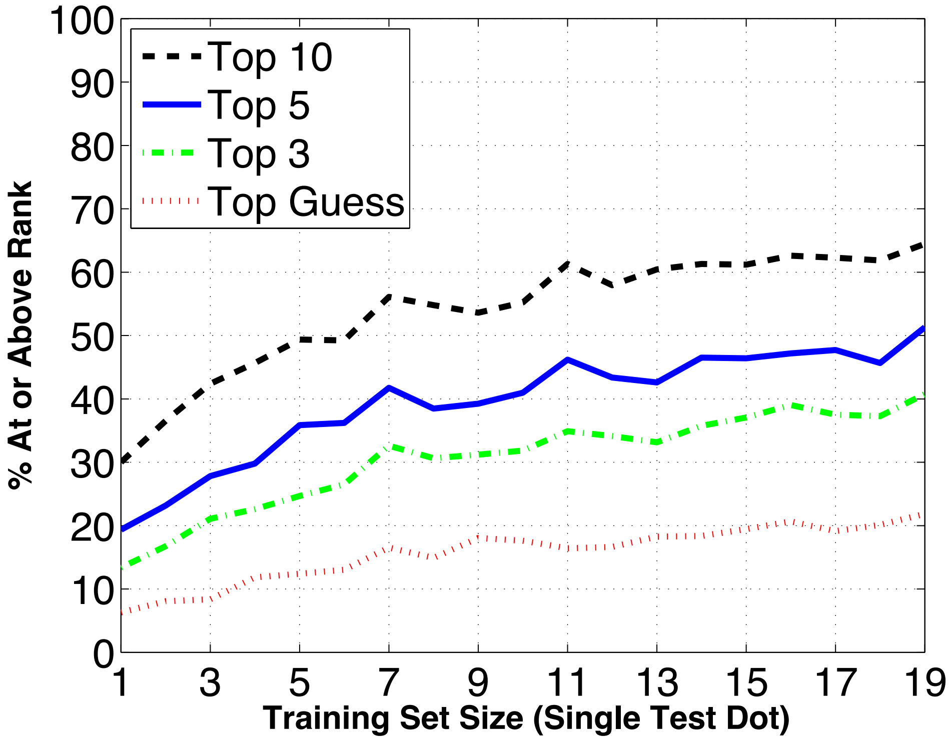


Figure 5.10: Increasing the training set size from 1 to 19 dots per respondent.

On average, our algorithm's first guess identified the correct respondent with 5.3% accuracy, five times better than the expected value for random guessing. See Figure 5.9, which shows the percentage of test bubbles for which the correct respondent fell at or above each possible rank. The correct respondent was in the top ten guesses 31.4% of the time. This result suggests that individuals can inadvertently convey information about their identities from even a single completed bubble.

Increasing training set size. In practice, respondents rarely fill out a single bubble on a form, and no two marked bubbles will be exactly the same. By training on multiple bubbles, we can isolate patterns that are consistent and distinguishing for a respondent from ones that are largely random. This experiment sought to verify this intuition by confirming that an increase in the number of training samples per respondent increases accuracy. We held our test set at a single bubble for each respondent and varied the training set size from 1 to 19 bubbles per respondent (recall that we have twenty total bubbles per respondent).

Figure 5.10 shows the impact various training set sizes had on whether the correct respondent was the top guess or fell in the top 3, 5, or 10 guesses. Given nineteen training dots and a single test dot, our first guess was correct 21.8% of the time. The graph demonstrates that a greater number of training examples tends to result in more accurate predictions, even with a single-dot test set. For the nineteen training dots case, the correct respondent was in the top 3 guesses 40.8% of the time and the top 10 guesses 64.5% of the time.

Increasing test set size. This experiment is similar to the previous experiment, but we instead held the training set at a single bubble per respondent and varied the test set size from 1 to 19 bubbles per respondent. Intuitively, increasing the number of examples per respondent in the test set helps ensure that our algorithms guess based on consistent features—even if the training set is a single noisy bubble.

Figure 5.11 shows the impact of various test set sizes on whether the correct respondent was the top guess or fell in the top 3, 5, or 10 guesses. We see more gradual improvements when increasing the test set size than observed when increasing training set size in the previous test. From one to nineteen test bubbles per respondent, the accuracy of our top 3 and 5 guesses increases relatively linearly with test set size, yielding maximum improvements of 4.3% and 7.6% respectively. For the top-guess case, accuracy increases with test set size from 5.3% at one bubble per respondent to 8.1% at eight bubbles then roughly plateaus. Similarly, the top 10 guesses case plateaus near ten bubbles and has a maximum improvement of 8.0%. Starting from

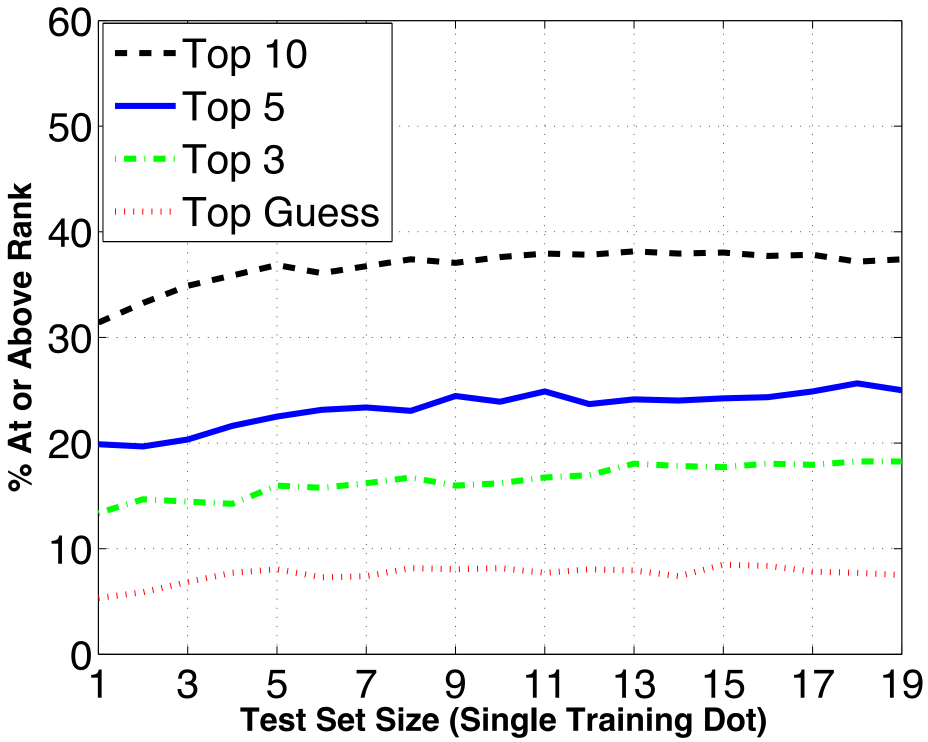


Figure 5.11: Increasing the test set size from 1 to 19 dots per respondent.

equivalent sizes, the marginal returns from increasing the training set size generally exceed those seen as test set size increases. Next, we explore the tradeoff between both set sizes given a fixed total of twenty bubbles per respondent.

Training-test set size tradeoff. Because we have a constraint of twenty bubbles per sample respondent, the combined total size of our training and test sets per respondent is limited to twenty. This experiment examined the tradeoff between the sizes of these sets. For each value of x from 1 to 19, we set the size of the training set per respondent to x and the test set size to $20 - x$. In some scenarios, a person analyzing bubbles would have far larger training and test sets than in this experiment. Fortunately, having more bubbles would not harm performance: an analyst could always choose a subset of the bubbles if it did. Therefore, our results provide a lower bound for these scenarios.

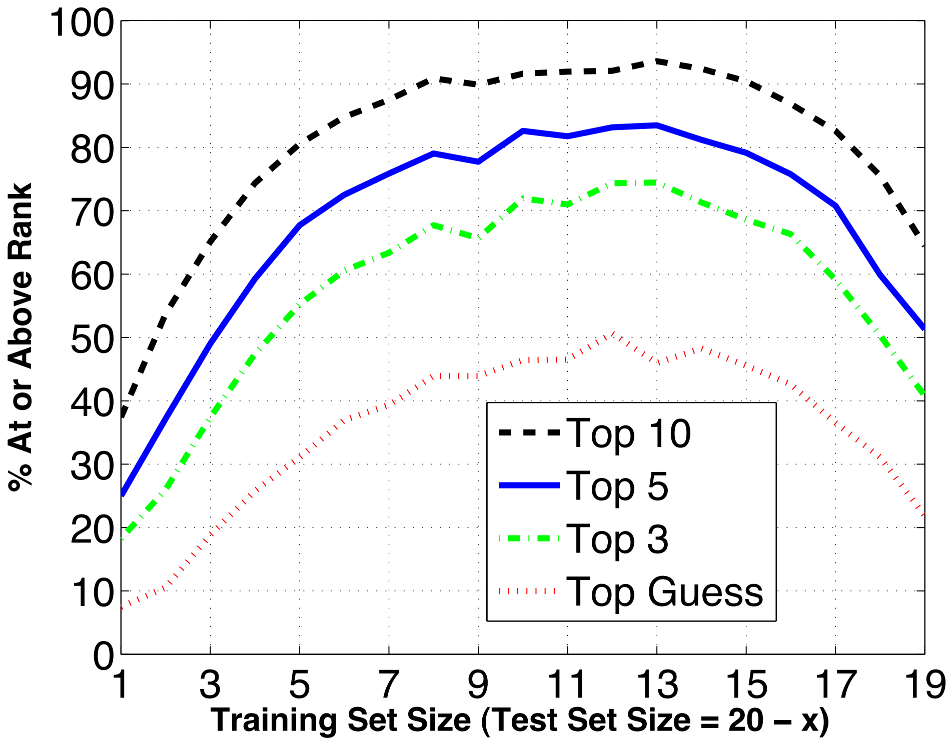


Figure 5.12: Trade-off between training and test set sizes.

Figure 5.12 shows how varying training/test set sizes affected whether the correct respondent was the top guess or fell in the top 3, 5, or 10 guesses. As the graph demonstrates, the optimal tradeoff was achieved with roughly twelve bubbles per respondent in the training set and eight bubbles per respondent in the test set.

Validation with gray bubbles. To further validate our methods, we tested the accuracy of our algorithms with a set of bubbles that we previously excluded: bubbles with gray backgrounds. These bubbles pose a significant challenge as the paper has both a grayish hue and a regular pattern of darker spots. This not only makes it harder to distinguish between gray pencil lead and the paper background but also limits differences in color distribution between users. See Figure 5.13.

As before, we selected surveys by locating ones with five completed (gray) bubbles for each answer choice, 1-4, yielding 97 surveys. We use twelve bubbles per respondent

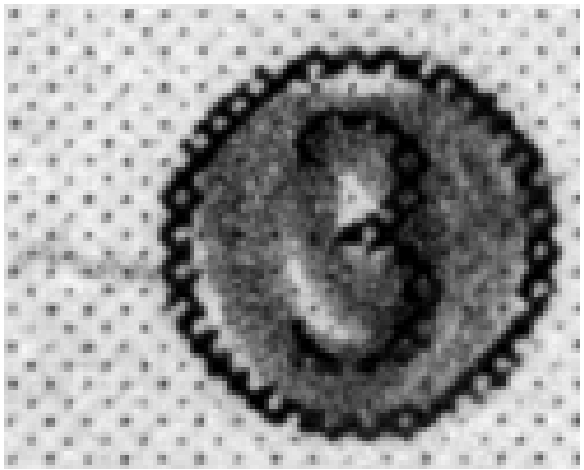


Figure 5.13: This respondent tends to have a circular pattern with a flourish stroke at the end. The gray background makes the flourish stroke harder to detect programatically. There isn't sufficient contrast – in general – to detect the flourish consistently.

in the training set and eight bubbles in the test set, and we apply the same algorithms and parameters for this test as the test in Section 5.2.1 on a white background.

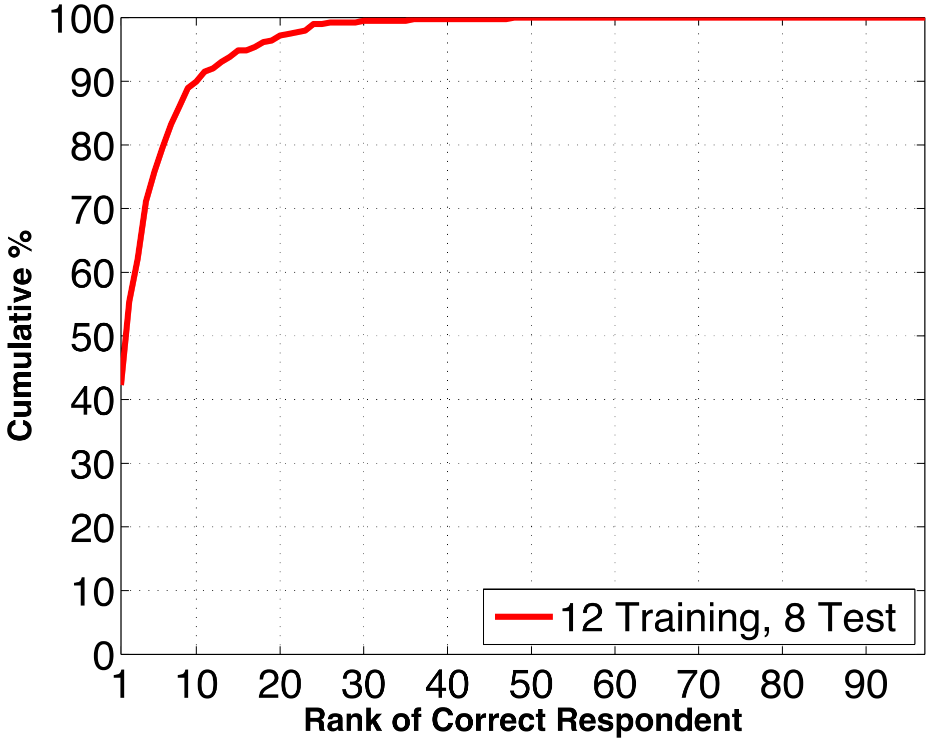


Figure 5.14: Using the unmodified algorithm with the same configuration as in Figure 5.6 on dots with gray backgrounds, we see only a mild decrease in accuracy.

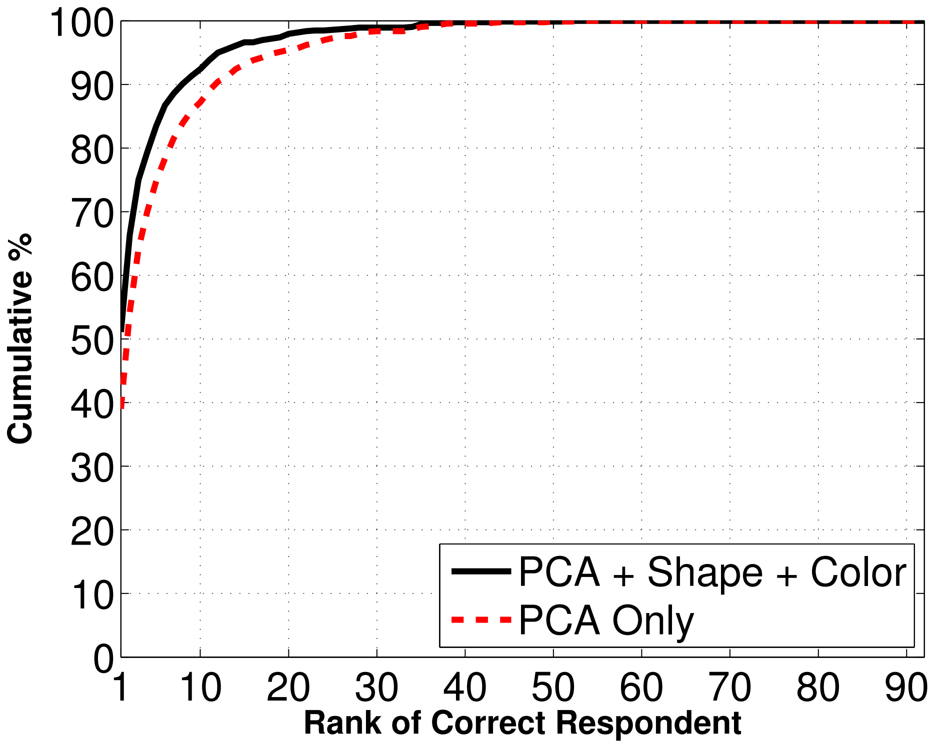
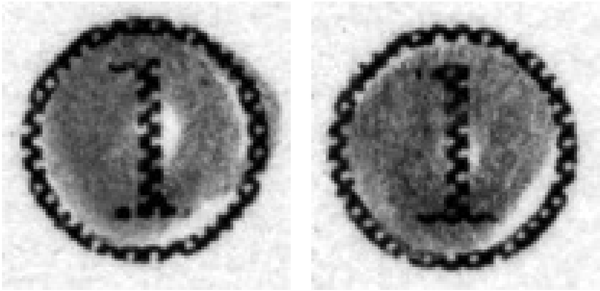


Figure 5.15: Performance with various combinations of features.

Figure 5.14 shows the percentage of test cases for which the correct respondent fell at or above each possible rank. Our first guess is correct 42.3% of the time, with the correct respondent falling in the top 3, 5, and 10 guesses 62.1%, 75.8%, and 90.0% of the time respectively. While slightly weaker than the results on a white background for reasons specified above, this experiment suggests that our strong results are not simply a byproduct of our initial dataset.

Feature vector options. As discussed in Section 5.1.1, our feature vectors combine PCA data, shape descriptors, and a custom color distribution to compensate for the limited data available from bubble markings. We tested the performance of our algorithms for equivalent parameters with PCA alone and with all three features combined. This test ran under the same setup as Figure 5.6 in Section 5.2.1.



(a) Person A (b) Person B

Figure 5.16: Bubbles from respondents often mistaken for each other. Both respondents use superficially similar techniques, leaving unmarked space in similar locations.

For both PCA and the full feature set, Figure 5.15 shows the percentage of test cases for which the correct respondent fell at or above each possible rank. The additional features improve the accuracy of our algorithm's first guess from 39.0% to 51.1% and the accuracy of the top ten guesses from 87.2% to 92.4%.

5.2.4 Discussion

Although our accuracy exceeds 50% for respondent re-identification, the restrictive nature of marking a bubble limits the distinguishability between users. We briefly consider a challenging case here.

Figure 5.16 shows marked bubbles from two respondents that our algorithm often mistakes for one another. Both individuals appear to use similar techniques to complete a bubble: a circular motion that seldom deviates from the circle boundary, leaving white-space both in the center and at similar locations near the border. Unless the minor differences between these bubbles are consistently demonstrated by the corresponding respondents, differentiating between these cases could prove quite difficult. The task of completing a bubble is constrained enough that close cases are nearly inevitable. In spite of these challenges, however, re-identification and detection of unauthorized respondents are feasible in practice.

5.3 Impact

This work has both positive and negative implications depending on the context and application. While we limit our discussion to standardized tests, elections, surveys, and authentication, the ability to derive distinctive bubble completion patterns for individuals may have consequences beyond those examined here. In Section 5.6, we discuss additional tests that would allow us to better assess the impact in several of these scenarios. In particular, most of these cases assume that an individual's distinguishing features remain relatively stable over time. Tests on longitudinal data are necessary to evaluate this assumption.

5.3.1 Standardized Tests

Scores on standardized tests may affect academic progress, job prospects, educator advancement, and school funding, among other possibilities. These high stakes provide an incentive for numerous parties to cheat and for numerous other parties to ensure the validity of the results. In certain cheating scenarios, another party answers questions on behalf of an official test-taker. For example, a surrogate could perform the entire test, or a proctor could change answer sheets after the test [41,58]. The recent cheating scandal in Long Island, NY, where over 20 students are accused of cheating, is one of a number of recent examples where cheating was widespread [7]. The ability to detect when someone other than the authorized test-taker completes some or all of the answers on a bubble form could help deter this form of cheating.

Depending on the specific threat and available data, several uses of our techniques exist. Given past answer sheets, test registration forms, or other bubbles ostensibly from the same test-takers, we could train a model as in Section 5.2.2 and use it to infer whether a surrogate completed some or all of a test.[6] Although the surrogate

[6] Note our assumption that the same unauthorized individual has not completed both the training bubbles and the current answer sheet.

may not be in the training set, we may rely on the fact that the surrogate is less likely to have bubble patterns similar to the authorized test-taker than to another set of test-takers. Because our techniques are automated, they could flag the most anomalous cases—i.e., the cases that would be rejected even under the least restrictive thresholds—in large-scale datasets for manual review.

If concern exists that someone changed certain answers after the test (for example, a proctor corrected the first five answers for all tests), we could search for questions that are correctly answered at an usually high rate. Given this information, two possible analysis techniques exists. First, we could train on the less suspicious questions, e.g. incorrect answer selections, and use the techniques of Section 5.2.2 to determine whether the suspicious ones on a form are from the same test-taker. In the case where a proctor may modify the markings for wrong *and* right answers, a question can be categorized as suspicious if it contains erasure markings [83]. Alternatively, we could train on the non-suspicious answer choices from each form and the suspicious answer choices from *all* forms other than a form of interest. Given this model, we could apply the techniques of Section 5.2.1 to see whether suspicious bubbles on that form more closely match less-suspicious bubbles on the same form or suspicious bubbles on other forms.

5.3.2 Elections

Our techniques provide a powerful tool for detecting certain forms of election fraud but also pose a threat to voter privacy.

Bubble-based analysis can help uncover fraudulent absentee ballots. Because absentee ballots do not require a voter to be physically present, concerns exist about individuals fraudulently obtaining and submitting these ballots [64]. For example, nursing home operators are regularly suspected of coercing the votes of their patients [64]. By identifying patterns in bubbles across absentee ballots submitted from the same address, election officials could gain confidence that the ballot was filled out by the intended party.

Unfortunately, because bubble markings can serve as a biometric, they can also be used in combination with seemingly innocuous auxiliary data to undermine ballot secrecy. Some jurisdictions now release scanned images of ballots following elections with the goal of increasing transparency (e.g., Humboldt County, California [1], which releases ballot scans at 300 DPI). If someone has access to these images or otherwise has the ability to obtain ballot scans, they can attempt to undermine voter privacy. Although elections may be decided by millions of voters, an attacker could focus exclusively on ballots cast in a target's precinct. New Jersey readjusts larger election districts to contain fewer than 750 registered voters [2]. Assuming 50% turnout, ballots in these districts would fall in groups of 375 or smaller. In Wisconsin's contentious 2011 State Supreme Court race, 71% of reported votes cast fell in the 91% of Wisconsin wards with 1,000 or fewer total votes [103].

Suppose that an interested party, such as a potential employer, wishes to determine how you voted. Given the ability to obtain bubble markings known to be from you (for example, on an employment application), that party can replicate our experiment in Section 5.2.1 to isolate one or a small subset of potential corresponding ballots. What makes this breach of privacy troubling is that it occurs without the consent of the voter and requires no special access to the ballots (unlike paper fingerprinting techniques, which require access to the physical ballot). The voter has not attempted to make an identifying mark, but the act of voting results in identifying marks nonetheless. This threat exists not only in traditional government elections but also in union and other elections, assuming that cast ballots are published, or otherwise available to the attacker.

Finally, one known threat against voting systems is pattern voting. For this threat, an attacker coerces a voter to select a preferred option in a relevant race and an unusual combination of choices for the other races. The unusual voting pattern will allow the attacker to locate the ballot later and confirm that the voter selected the correct choice for the relevant race. One proposed solution for pattern voting is to cut ballots apart to separate votes in individual contests [24]. Our work raises the possibility that physically divided portions of a ballot could be connected, undermining this mitigation strategy.

5.3.3 Surveys

Human subjects research is governed by a variety of restrictions and best practices intended to protect subjects from harm. One factor to be considered when collecting certain forms of data is the level of anonymity afforded to subjects. If a dataset contains identifying information, such as subject name, this may impact the types of data that should be collected and procedural safeguards imposed to protect subject privacy. If subjects provide data using bubble forms, these markings might effectively serve as a form of identifying information, tying the form to the subject even in the absence of a name. Re-identification of subjects can proceed in the same manner as re-identification of voters, by matching marks from a known individual against completed surveys (as in Section 5.2.1).

Regardless of whether ethical or legal questions are raised by the ability to identify survey respondents, this ability might affect the honesty of respondents who are aware of the issue. Dishonesty also poses a problem for commercial surveys that do not adhere to the typical practices of human subjects research.

The impact of this work for surveys is not entirely negative, however. In certain scenarios, the person responsible for administering a survey may complete the forms herself or modify completed forms, whether to avoid the work of conducting the survey or to yield a desired outcome. Should this risk exist, similar analysis to the standardized test and election cases could help uncover the issue.

5.3.4 Authentication

Because bubble markings are a biometric, they may be used alone or in combination with other techniques for authentication. Using a finger or a stylus, an individual could fill in a bubble on a pad or a touchscreen. Because a computer could monitor user input, various details such as velocity and pressure could also be collected and used to increase the accuracy of identification, potentially achieving far stronger results than in Section 5.2.2. On touchscreen devices, this technique may or may not be easier for users than entry of numeric codes or passwords. Additional testing would be necessary for this application, including tests of its performance in the presence of persistent adversaries. While bubble markings alone are unlikely to provide very high confidence, they could be useful as part of a multi-factor authentication strategy.

5.4 Mitigation

The impact of this chapter's techniques can be both beneficial and detrimental, but the drawbacks may outweigh the benefits for certain applications. In those circumstances where bubble-based techniques are necessary, or in cases where access to marked bubbles is desirable for other reasons, a mitigation strategy is necessary. Depending on the application, the appropriate mitigation strategy varies. We discuss three classes of mitigation strategies. First, we examine procedural safeguards that restrict access to forms or scanned images. Second, we consider changes to the forms themselves or how individuals mark the forms. Finally, we explore techniques that obscure or remove identifying characteristics from scanned images. No strategy alone is perfect, but various combinations may be acceptable under different circumstances.

Elections are one application where a mitigation strategy is necessary. The increasing focus on the fairness and transparency of elections has prompted a number of different strategies designed to achieve greater transparency. One approach spearheaded by Humboldt County, located in northern California, is to publish completed ballots online after each election [43]. This enables individual citizens to perform a manual recount, verifying the official results, hopefully leading to an increase in voter confidence. Interestingly, there are examples of significant errors being detected by members of the public based on these manual recounts [108]. In one such example, 197 votes were not included in the official tally [108]. The accidental exclusion of these votes was not detected by the audit mechanism in place at the time. Examples like this significantly decrease the likelihood that future access to completed ballots will be stymied. The long-term implication is that marked bubbles will be publicly available – potentially revealing information about how a particular citizen voted. This rest of this section focuses on a variety of strategies to mitigate some of the risks associated with many of the previously discussed applications, but with a primary focus on mitigation strategies for elections.

5.4.1 Procedural Safeguards

Procedural safeguards that restrict access to both forms themselves and scanned images can be both straightforward and effective. Collection of data from bubble forms typically relies on scanning the forms, but a scanner need not retain image data for any longer than required to process a respondent's choices. If the form and its image are unavailable to an adversary, our techniques would be infeasible.

In some cases, instructive practices or alternative techniques already exist. For example, researchers conducting surveys could treat forms with bubble markings in the same manner as they would treat other forms containing identifying information. In the context of elections, some jurisdictions currently release scanned ballot images following an election to provide a measure of transparency. This release is not a satisfactory replacement for a statistically significant manual audit of paper ballots (e.g., [10,21]), and a public release is not necessary for such an audit. Because scanned images could be manipulated or replaced, statistically significant manual confirmation of the reported ballots' validity remains necessary. Furthermore, releasing the recorded choices from a ballot (e.g., Washington selected for President, Lincoln selected for Senator, etc.) without a scanned ballot image is sufficient for a manual audit.

Whether the perceived transparency provided by the release of ballot scans justifies the resulting privacy risk is outside the scope of this work. Nevertheless, should the release of scanned images be desirable, the next section describes methods to change the form or marking device to minimize variation in markings between respondents.

5.4.2 Changes to Forms or Marking Devices

As explored in Section 5.2.3, changes to the forms themselves such as a gray background can impact the accuracy of our tests. The unintentional protection provided by this particular change was mild and unlikely to be insurmountable. Nevertheless, more dramatic changes to either the forms themselves or the ways people mark them could provide a greater measure of defense.

Changes to the forms themselves should strive to limit either the space for observable human variation or the ability of an analyst to perceive these variations. The addition of a random speckled or striped background in the same color as the writing instrument could create difficulties in cleanly identifying and matching a mark. If bubbles had wider borders, respondents would be less likely to color outside the lines, decreasing this source of information. Bubbles of different shapes or alternate marking techniques could encourage less variation between users. For example, some optical scan ballots require a voter simply to draw a line to complete an arrow shape [5], and these lines may provide less identifying information than a completed bubble.

The marking instruments that individuals use could also help leak less identifying information. Some Los Angeles County voters use ink-marking devices, which stamp a circle of ink for a user [59]. Use of an ink-stamper would reduce the distinguishability of markings, and even a wide marker could reduce the space for inadvertent variation.

5.4.3 Scrubbing Scanned Images

In some cases, the release of scanned bubble forms themselves might be desirable. In California, Humboldt County's release of ballot image scans following the 2008 election uncovered evidence of a software glitch causing certain ballots to be ignored [44]. Although a full manual audit could have caught this error with high probability, ballot images provide some protection against unintentional errors in the absence of such audits.

In this case, the ability to remove identifying information from scanned forms while retaining some evidence of a respondent's actions is desirable. One straightforward approach is to cover the respondent's recorded choices with solid black circles. Barring any stray marks or misreadings, this choice would completely remove all identifying bubble patterns. Unfortunately, this approach has several disadvantages. First, a circle could cover choices that were not selected, hiding certain forms of errors. Second, suppose that a bubble is marked but not recorded. While the resulting image would allow reviewers to uncover the error, such marks retain a respondent's identifying details. The threat of a misreading and re-identification could be sufficient to undermine respondent confidence, enabling coercion.

An alternative to the use of black circles is to replace the contents of each bubble with its average color, whether the respondent is or is not believed to have selected the bubble. The rest of the scan could be scrubbed of stray marks. This would reduce the space for variation to color and pressure properties alone. Unfortunately, no evidence exists that these properties cannot still be distinguishing. In addition, an average might remove a respondent's intent, even when that intent may have been clear to the scanner interpreting the original form. Similar mitigation techniques involve blurring the image, reducing the image resolution, or making the image strictly black and white, all of which have similar disadvantages to averaging colors.

One interesting approach comes from the facial image recognition community. Newton et al. [74] describe a method for generating k-anonymous facial images. This technique replaces each face with a "distorted" image that is k-anonymous with respect to faces in the input set. The resulting k-anonymous image maintains the expected visual appearance, that of a human face. The exact details are beyond the scope of this thesis, but the underlying technique reduces the dimensionality using Principal Component Analysis and an algorithm for removing the most distinctive features of each face [74].

Application of facial anonymization to bubbles is straightforward. Taking marked and unmarked bubbles from all ballots in a set, we can apply the techniques of Newton et al. to each bubble, replacing it with its k-anonymous counterpart. The result would roughly maintain the visual appearance of each bubble while removing certain unique attributes. Unfortunately, this approach is imperfect in this scenario. Replacement of an image might hide a respondent's otherwise obvious intent. In addition, distinguishing trends might occur over multiple bubbles on a form: for example, an individual might mark bubbles differently near the end of forms (this is also a problem for averaging the bubble colors). Finally, concerns exist over the guarantees provided by k-anonymity [15]. For example, it does not follow that given a set of n *individually* k-anonymous bubbles that the set of k-anonymous bubbles are also k-anonymous. In contrast, differential privacy [35]provides guarantees on this type of composition, i.e. a set of n $\epsilon-$differentially private images would be $\epsilon n-$differentially private [67]. Future work is necessary to explore the application of differential privacy to images.

We caution that the value of these images for proving the true contents of physical bubble forms is limited: an adversary with access to the images, whether scrubbed or not, could intentionally modify them to match a desired result. These approaches are most useful where the primary concern is unintentional error.

5.5 Related Work

There are a number of existing techniques which address applications related to those presented in this chapter. It is informative to compare and contrast these with the bubble-based techniques. In practice, bubble-based techniques are often complementary to existing approaches, permitting outcomes which were not originally achievable.

Biometrics. Biometrics can be based on physical or behavioral characteristics of an individual. Physical biometrics are based on physical characteristics of a person, such as fingerprints, facial features, and iris patterns. Behavioral biometrics are based on behavioral characteristics that tend to be stable and difficult to replicate, such as speech or handwriting/signature [50]. Bubble completion patterns are a form of behavioral biometric.

As a biometric, bubble completion patterns are similar to handwriting, though handwriting tends to rely on a richer, less constrained set of available features. In either case, analysis may occur on-line or off-line [71]. In an on-line process, the verifying party may monitor characteristics like stroke speed and pressure. In an off-line process, a verifying party only receives the resulting data, such as a completed bubble. Handwriting-based recognition sometimes occurs in an on-line setting. Because off-line recognition is more generally applicable, our analysis occurred purely in an off-line manner. In some settings, such as authentication, on-line recognition would be possible and could yield stronger results.

Document re-identification. Some work seeks to re-identify a precise physical document for forgery and counterfeiting detection (e.g., [26]). While the presence of biometrics may assist in re-identification, the problems discussed in this work differ. We seek to discover whether sets of marked bubbles were produced by the same individual. Our work is agnostic with respect to the origin of the sets, i.e. whether they come from the same form, different forms, or duplicates of forms. Nevertheless, our work and document re-identification provide complementary techniques. For example, document re-identification could help determine whether the bubble form (ballot, answer sheet, survey, etc.) provided to an individual matches the one returned or detect the presence of fraudulently added forms.

Cheating Detection. Existing work uses patterns in answer choices themselves as evidence of cheating. Myagkov et al. uncover indicators of election fraud using aggregate vote tallies, turnout, and historical data [70]. Similarly, analysis of answers on standardized tests can be particularly useful in uncovering cheating [40, 58]. For example, if students in a class demonstrate mediocre overall performance on a test yet all correctly answer a series of difficult questions, this may raise concerns of cheating. The general strategy in this line of research is to look for answers that are anomalous in the context of either other answers or auxiliary data.

Bubble-based analysis is also complementary to these anti-cheating measures. Each technique effectively isolates sets of suspicious forms, and the combination of the two would likely be more accurate than each independently. Although our techniques alone do not exploit contextual data, they have the advantage of being unbiased by that data. If a student dramatically improves her study habits, the resulting improvement in test scores alone might be flagged by other anti-cheating measures but not our techniques.

5.6 Future Work

Although a variety of avenues for future work exist, we focus primarily on possibilities for additional testing and application-specific uses here.

Our sample surveys allowed a diverse set of tests, but access to different datasets would enable additional useful tests. We are particularly interested in obtaining and using longitudinal studies—in which a common set of respondents fill out bubble forms multiple times over some period—to evaluate our methods. While providing an increased number of examples, this could also identify how a respondent's markings vary over time, establish consistency over longer durations, and confirm that our results are not significantly impacted by writing utensil. Because bubble forms from longitudinal studies are not widely available, this might entail collecting the data ourselves.

While we tested our techniques using circular bubbles with numbers inscribed, numerous other form styles exist. In some cases, respondents instead fill in ovals or rectangles. In other cases, selection differs dramatically from the traditional fill-in-the-shape approach—for example, the line-drawing approach discussed in Section 5.4 bears little similarity to our sample forms. Testing these cases would not only explore the limits of our work but could also help uncover mitigation strategies.

Section 5.3 discusses a number of applications of our techniques. Adapting the techniques to work in these scenarios is not always trivial. For example, Section 5.5 discusses existing anti-cheating techniques for standardized tests. Combining the evidence provided by existing techniques and ours would strengthen anti-cheating measures, but it would also require some care to process the data quickly and merge results.

Use of bubble markings for authentication would require both additional testing and additional refinement of our techniques. Given a dataset containing on-line information, such as writing instrument position, velocity, and pressure, we could add this data to our feature vectors and test the accuracy of our techniques with these new features. This additional information could increase identifiability considerably—signature verification is commonly done on-line due to the utility of this data—and may yield an effective authentication system. Depending on the application, a bubble-based authentication system would potentially need to work with a finger rather than a pen or stylus. Because the task of filling in a bubble is relatively constrained, this application would require cautious testing to ensure that an adversary cannot impersonate a legitimate user.

5.7 Discussion

Marking a bubble is an extremely narrow task, but as this work illustrates, the task provides sufficient expressive power for individuals to unintentionally distinguish themselves. Using a dataset with 92 individuals, we demonstrate how to re-identify a respondent's survey with over 51% accuracy. In addition, we are able to detect an unauthorized respondent with over 92% accuracy with a false positive rate below 10%. We achieve these results while performing off-line analysis exclusively, but on-line analysis has the potential to achieve even higher rates of accuracy.

The implications of this study extend to any system utilizing bubble forms to obtain user input, especially cases for which protection or confirmation of a respondent's identity is important. Additional tests can better establish the threat (or benefit) posed in real-world scenarios. Mitigating the amount of information conveyed through marked bubbles is an open problem, and solutions are dependent on the application. For privacy-critical applications, such the publication of ballots, we suggest that groups releasing data consider means of masking respondents' markings prior to publication.

Chapter 6

Identifying LoudSpeakers

Loudspeakers, more commonly known as audio speakers or just speakers, are ubiquitous in our daily lives. Televisions, stereo systems, cars, phones, and computers (to name just a few devices) all have loudspeakers, and most of us hear loudspeaker-generated sounds every day. In their traditional role, loudspeakers are treated as simple devices that reproduce an input audio signal, e.g. a movie soundtrack or music. It may be surprising that loudspeakers, even those of the same make and model, reproduce the same audio signal differently. Our experiments show that each loudspeaker generates a characteristic distortion of the input signal. These distortions are measurable using only a cheap microphone and are consistently reproduced, allowing individual loudspeakers to be distinguished from one another.

When playing the same sound through speakers of different make or model, we expect to hear slightly different outputs from each speaker. One speaker may be louder, softer or 'richer' for a given audio signal. These perceived differences are the result of varying designs, materials and manufacturing processes. Listeners understand that audio generated by different models of loudspeakers will generate different output. When purchasing loudspeakers, buyers select the make or model that sounds the best, given their price range. However, when dealing with multiple instances of a loudspeaker, those of the same make and model, listeners (falsely) expect identical outputs. Purchasers do not often request the particular speaker on which they based their selection; they are satisfied with one of the same make and model.

We perform a series of experiments to measure the characteristic distortion of loudspeakers and the ability to distinguish between them based on this distortion. By combining signal processing techniques and machine learning, we are able to distinguish among a set of 19 seemingly identical loudspeakers of the same make and model with over 98% accuracy. Using Amazon's Mechanical Turk, we obtained measurements from 107 loudspeakers of different makes and models. From this set, we are able to re-identify individual loudspeakers with over 81% accuracy. For some applications, distinguishing between a known set is not enough. We also explore the ability to fingerprint loudspeakers, producing an equal-error-rate of approximately 15% based on measurements from 107 different loudspeakers. We also experimented with different types of input signals and the resiliency of loudspeaker identification to noise.

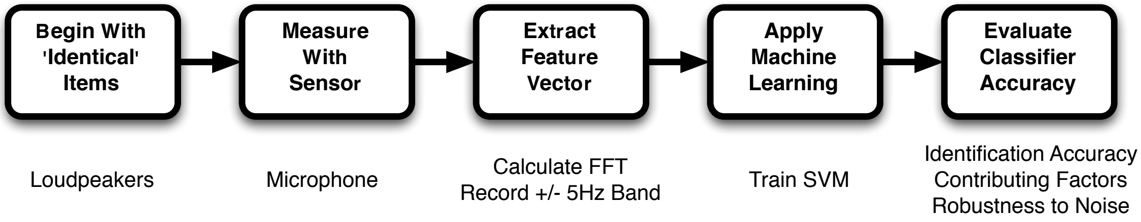


Figure 6.1: Application of the General Framework, introduced in Chapter 2, to distinguishing between seemingly identical loudspeakers.

To put this work in context, we briefly discuss some related techniques, e.g. the ability to identify speakers (as opposed to loudspeakers), and relevant signal processing techniques in Section 6.1. The remainder of this chapter closely follows the General Framework, see Figure 6.1. Section 6.2 (Stage 1), provides an overview of how loudspeakers work and the components that contribute to the measured distortions. Next, we describe how to measure the distortions generated by a loudspeaker and how to generate a feature vector that captures this characteristic distortion in Section 6.3 (Stages 2-3). In Section 6.3.3 (Stage 4), we apply machine learning to train a classifier to distinguish between feature vectors originating from different loudspeakers. In a series of experiments we evaluate the ability of our classifier to distinguish between loudspeakers under a variety of conditions in Section 6.4 (Stage 5). A number of factors can contribute to the measured distortions and we perform a series of experiments, described in Section 6.5, that investigate their impact on the ability of loudspeakers to be identified. In Section 6.6 we explore the ability to authenticate a particular loudspeaker and in Section 6.7 we evaluate the potential for forging a loudspeaker's distortion. We then describe a number of applications and implications for this technique in Sections 6.8 and 6.9. Finally, we summarize our conclusions in 6.10.

6.1 Background and Related Work

Distinguishing other devices. Identifying physical devices based on deviations from expected properties is not a new concept. In recent years the research community has developed a variety of methods to identify scanners, digital cameras, RFID tags and distinguish between various appliances [37, 57, 61, 107]. In each of these cases, manufacturing defects and inherent physical attributes of embedded components make devices in each of these categories identifiable.

Desktop scanners are identifiable based on the noise pattern of their particular light sensors [57]. Due to variations in the sensitivity of the light sensor on scanners, scanned images exhibit a characteristic scanner sensor noise. By measuring the noise signature of a scanned image, the scanner on which an image was scanned can be identified with high accuracy. Digital cameras exhibit similar noise signatures. With digital cameras, the sensor noise is the result of non-uniformity in the photoresponse of their light sensors [61].

More recently, Gupta et. al. explore the implications of Electro-Magnetic Interference (EMI) caused by switched mode power supplies. The power supplies in many consumer devices induce a characteristic noise pattern on the power lines. Gupta et.al. show that this noise pattern can be measured via the home's power lines allowing devices, such as fluorescent bulbs, washing machines, laptop chargers, and televisions to be distinguished. Enev et al. explore the implications of this technique to distinguish between brands and models of televisions, and even detect the content displayed on the television [37].

Audio Fingerprinting. A technique related to the one we describe here is audio fingerprinting. The most well-known example is Shazam. Shazam is a commercial service that attempts to identify the song, or movie based on qualities of a recorded audio signal [34,102]. In general, audio fingerprinting services treat noise as a problem to overcome. In practice, audio fingerprinting services must differentiate between the audio signal and background noise as well has handle variations in an audio signal based on the channel encoding to identify the source audio signal. Our application is distinct from that of audio fingerprinting services. In the applications we are concerned with, we know the source audio signal and treat the 'noise'—deviations from the expected value—as the feature under study.

Speaker Identification. Speaker Identification, while similar in terminology, is a conceptually separate topic [84]. Speaker identification focuses on identifying which person is speaking in an audio recording. In this work, we are not focused on the individual who is speaking, but on identifying the loudspeaker that is generating a particular sound, and measuring its characteristic distortion.

Signal Processing Overview. Audio signals can be modeled as an infinite sum of sinusoidal waves. The Fourier Transform is a common technique in signal processing to determine the constituent sinusoidal components of an audio signal [78] The Fast Fourier Transform (FFT) simply measures the amplitude of each sine wave. In digital signals, the FFT is a finite sum that approximates the contribution of groups of frequencies. In our analysis, we made use of the FFTW library for calculating the Fast Fourier Transform of an audio signal [39]. The Fourier Transform measures both phase and amplitude of constituent frequencies. In this work, we focus on the amplitude.

6.2 Speaker Overview

Loudspeakers from different manufacturers or different models from the same manufacturer have different designs and are often crafted out of different materials. To generate sound from an input signal, loudspeakers rely on the same basic physical mechanism. In this section, we describe at a high level how loudspeakers work and how the physical components of loudspeakers contribute to measurable distortions in the generated audio signal.

6.2.1 How a Speaker Works

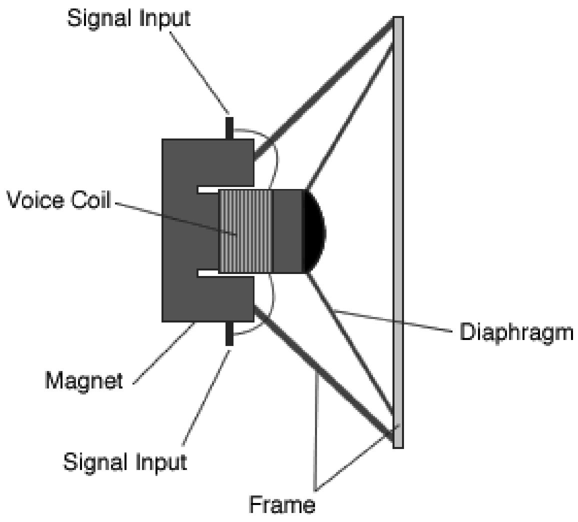


Figure 6.2: Cross section of a typical loudspeaker. The main component is the circular magnet, which creates a constant magnetic field. Attached to the diaphragm is the voice coil also known as the driver. The driver is connected to amplifier. Changes in current induce a change in the magnetic field, causing the driver to move in response, moving the diaphragm, pushing the air. Image source [3].

Loudspeakers generate sound by creating pressure variations in a compressible medium, like air. These pressure variations are transferred through the air and interpreted by the ear, which measures changes in air pressure, as sound. The basic design of a loudspeaker is shown in Figure 6.2. A loudspeaker consists of a semi-rigid mesh membrane called the diaphragm that is attached to the loudspeaker driver. The driver moves the membrane forward and backward in proportion to the connected audio signal. The driver typically consists of an iron metal bar wrapped with a coil. The driver, also called the voice coil, is placed inside of a circular magnet, which creates a fixed magnetic field. To move the speaker, electrical current is passed through the metal coil causing the driver to act as an electromagnet. Changes in electrical current induce a change in the magnetic field causing the driver to physically move in response. This in turn moves the diaphragm, creating variations in air pressure, generating sound.

Each loudspeaker component can introduce variations into the generated sound. For example, variations in the electromagnetic properties of the driver can cause differences in the velocity and smoothness of motion when electrical current is applied. The diaphragm is a semi-rigid membrane whose material can differ based on the desired auditory properties. Defects in manufacturing of the membrane can contribute to deviations in the accurate reproduction of certain frequencies. Audiophiles speak about the burn-in effect as a method of breaking in a new loudspeaker. We do not measure the burn-in effect, but focus on the distinct variations produced by 'burned-in' loudspeakers. Resonant frequencies and manufacturing defects in any component may also play a role in the accurate reproduction of the audio signal [65].

6.2.2 The Frequency Response of a Speaker

A loudspeaker's frequency response is one common way to characterize a loudspeaker's ability to reproduce audio signals composed of a range of frequencies. An ideal loud-

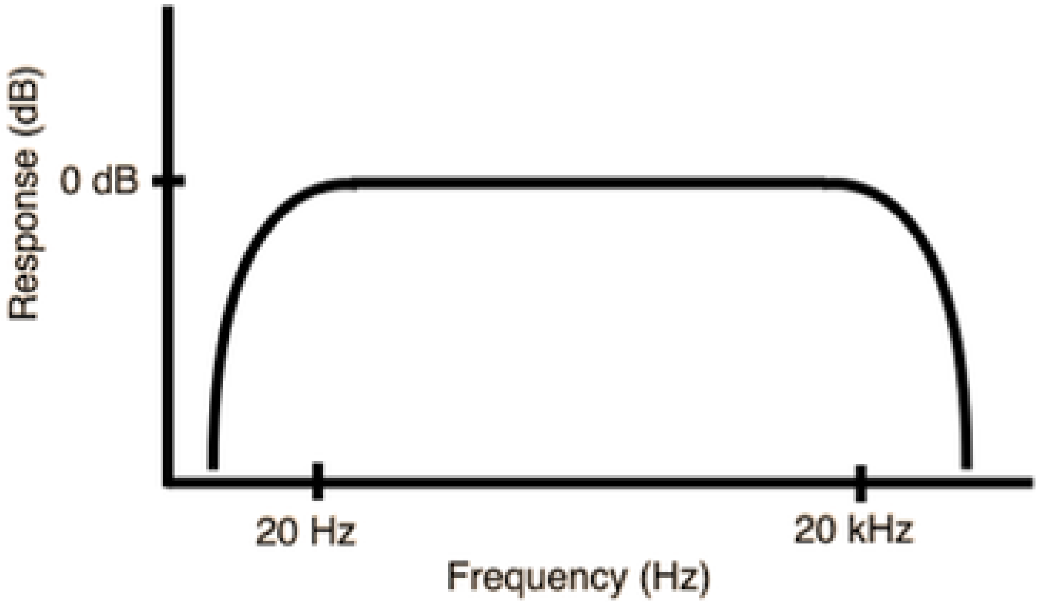


Figure 6.3: An Ideal speaker has a flat frequency response. A speaker with a flat response produces all frequencies with the same magnitude. In this context, the 0 dB line denotes the perfect reproduction of an input signal. Responses above 0 dB indicate amplification of a frequency, while responses below 0 dB indicate attenuation.

speaker reproduces all frequencies between 20 Hz to 20 kHz, roughly the range of human hearing, without amplifying or attenuating particular frequencies. For example, an ideal speaker could generate a 440 Hz tone with the same magnitude as a 4400 Hz tone. If a loudspeaker reproduces certain frequencies more strongly than others, the output will not be an accurate reproduction of the input audio signal.

The Ideal Speaker

A speaker that can reproduce frequencies between 20 Hz and 20 kHz with the same magnitude has a flat frequency response. See Figure 6.3. This range of frequencies is a common rule of thumb for the range of speakers that humans can hear [77]. A flat response means that the input signal is reproduced accurately, i.e. no distortion. If such a speaker were to exist, it would perfectly reproduce any input audio signal with frequencies within the flat region.

In practice, no loudspeaker has a perfectly flat frequency response across the range of desired frequencies. It is inherently difficult to manufacture a loudspeaker capable of generating a low-rumble with the same magnitude as a high-frequency tone. There is a physical limit regarding the range of frequencies that can be accurately reproduced by a speaker of a given size [69].

Real-World Speakers

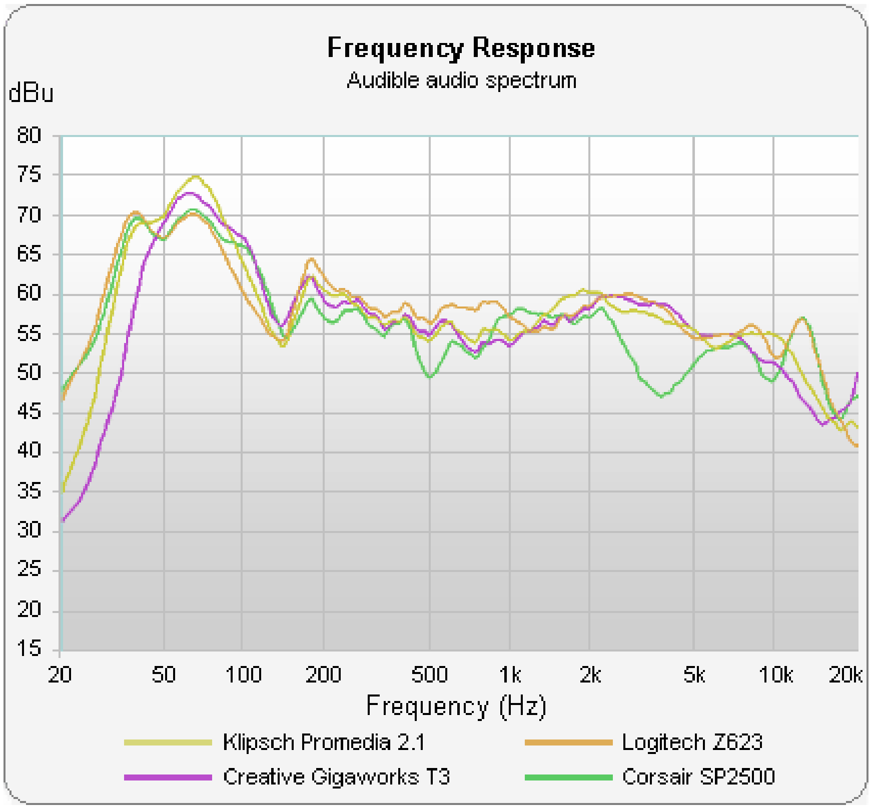


Figure 6.4: Frequency responses for 4 real speakers of different make and model. Source [104]

Each make and model of speaker has a different frequency response. The frequency responses of some real-world speakers are shown in Figure 6.4. Each loudspeaker generates the same input audio signal slightly differently, analogous to a built-in equalizer. For example, each of the measured loudspeakers generate bass frequencies (between 40-150 Hz) more strongly than higher frequencies. When compared to the other loudspeakers, the Corsair SP2500 generates a 3 kHz frequency with significantly less amplitude than a 5 kHz tone.

There is an implicit assumption that multiple instances of the same speaker will have identical frequency responses. This is often reinforced by the notion that any measured variations between audio generated by a speaker are random and non-reproducible across multiple recordings. If consistently measurable differences were present within a particular speaker model, tighter manufacturing tolerances would reduce these differences, resulting in a more consistent product.

We show that individual loudspeakers, even those of the same make and model, are identifiable based on measurable distortions. We measure these distortions using a simple sensor, a computer's built-in microphone. While microphones can introduce their own distortions into the recorded signal, we did not find this to be a significant contributing factor. We address this and other potential contributing factors in Section 6.5.

6.3 Measuring Speaker Distortion

To measure the distortion of each loudspeaker, we generate a series of tones spanning the range of frequencies that a loudspeaker is expected to reproduce. We generate a series of constant tones at frequencies between 300 Hz and 10 kHz at 150 Hz intervals. In total, 65 tones are generated. The output audio signal generated by each loudspeaker is recorded using a microphone as 16-bit Linear PCM at a sample rate of 44.1 kHz. This is equivalent to CD-ROM quality audio [88].

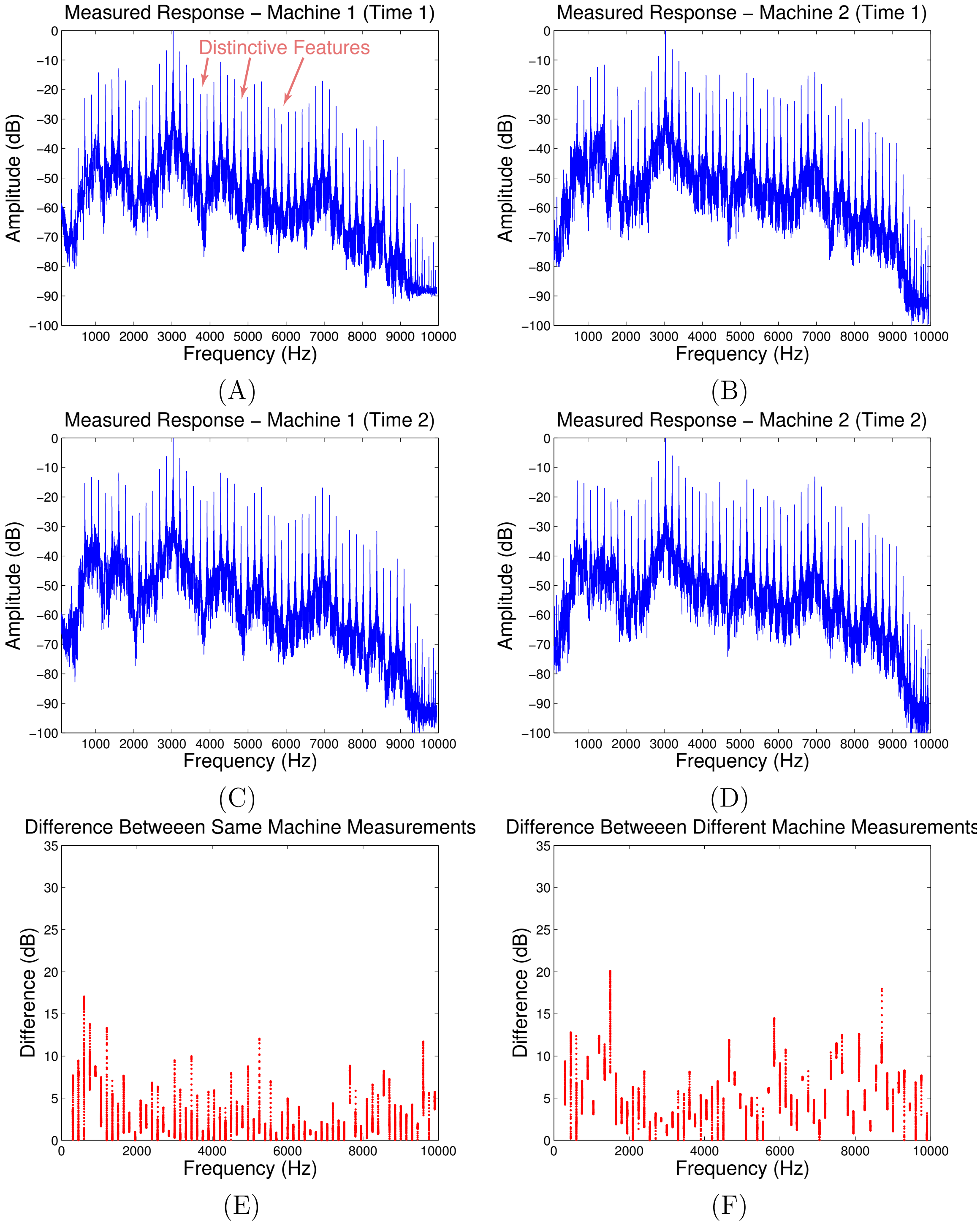
In principle, one could use the raw time-domain audio recording to distinguish between speakers. However, using the raw audio has several drawbacks. First, it makes identification difficult in the presence of noise. It is also inefficient to process such large files. Instead, we capture the unique properties of a loudspeaker by measuring the magnitudes of certain frequencies. We are particularly interested in the 65 frequencies in the input audio signal and their neighboring frequencies.

6.3.1 Distortion Characteristics

Loudspeakers vary dramatically in their ability to reproduce a wide range of frequencies. Figure 6.5 shows the FFT magnitude response for two of the laptop speakers in our experiments at two points in time. The input audio signal consisted of 65 frequencies with the same amplitude. Notice the wide variations in the peaks of the measured frequencies across the 300 Hz to 10 kHz range. In particular, notice the three dips in frequency magnitudes between 4000 Hz and 5000 Hz in measurements from Machine 1.

When reproducing an audio signal containing a set of constant tones, a loudspeaker varies not only in the magnitude of each target frequency, but also in the magnitude of frequencies near the target frequency. To measure the magnitude of frequencies in each recording, we took a single FFT of the recorded output. We generated a single FFT with a window size of 524288. The window is so large due to the fixed duration of our recordings, and zero-padding of the input signal to the nearest power of 2. We used a Blackman window half the size of the FFT window. Generating an FFT of this size results in a fine grained bin size of 0.084114 Hz. We experimentally found that performing a single FFT provided better results than a series of smaller FFTs. This may be due to distinguishing information being present during the transition between subsequently produced frequencies.

Figure 6.8 shows the distribution of frequencies around a target frequency of 1500 Hz for one of the speakers in our experiments. There is a wide spread of generated frequencies from the intended 1500 Hz tone. Generating a single FFT with such a large window can increase the resulting frequency spread, see Figure 6.8 [93]. Through experimentation on our set of homogeneous machines (Section 6.4.1), we found most



(A)

(B)

(C)

(D)

(E)

(F)

Figure 6.5: Frequency response measurements from two machines of the same make and model each playing the same series of 65 tones, recorded at two different times. There are noticeably distinct features in the responses of each machine, for example the dips in the response peaks highlighted in (A). The plot in (E) is absolute value of the difference between measurements from the same machine, $(E) = |(C) - (A)|$, for the 10Hz band surrounding each input tone (averaging 2.4 dB). In contrast, plot (F) shows this difference between the two machines (average 5.0 dB): $(F) = |(B) - (A)|$.

of the distinguishing information about a loudspeaker is captured in the 10 Hz band surrounding the target frequency. The remaining components are likely artifacts of the FFT.

Figure 6.5 (A-D) shows the FFT of measurements from two speakers at two points in time. Measurements from the same speaker look more similar to one another than measurements from different speakers. Figure 6.5A depicts the FFT of a measurement taken from a single loudspeaker that exhibits some visually distinctive features. These features also appear in Figure 6.5C, suggesting they are reproducible across measurements, providing a characteristic that can distinguish this loudspeaker. The feature vector for this loudspeaker should capture these and similarly distinctive characteristics.

6.3.2 Generating a Feature Vector

In principle, one could use the intensities of all of the frequency measurements as a feature vector. The resulting feature vector would be smaller than using the raw-audio signal, but is larger than necessary to distinguish between speakers. It is tempting to reduce the length of the feature vector by selecting the frequencies that seem to convey the most information. Many applications dealing with audio signal feature selection are interested in characterizing a song or movie based on differences in frequency composition. In our case, we are interested in the deviations of a particular loudspeaker, not a generic audio signal, and characterizing these variations in a concise manner.

In light of this, we generate a feature vector using a very simple method. We limit the frequencies in the feature vector to those nearest to the frequencies present in the input signal. We discard frequencies not sufficiently close to these expected frequencies. This has two main benefits. It reduces feature vector length and limits the impact of background noise, which, if present, is likely to be most prominent in the discarded frequencies.

To generate a feature vector for a loudspeaker, we play the same audio signal on each loudspeaker. The input audio signal consists of a series of 65 known frequencies. Figure 6.6 is the spectrogram of the raw input signal. We anticipate that each loudspeaker will generate sounds with frequencies near those intended. As previously mentioned, loudspeakers are not able to perfectly reproduce an input signal. This can be seen in the spectrogram of one of the recordings used the experiments in Figure 6.7. The frequency magnitudes generated by this loudspeaker vary, and it is unable to generate frequencies with significant magnitude above 8 kHz.

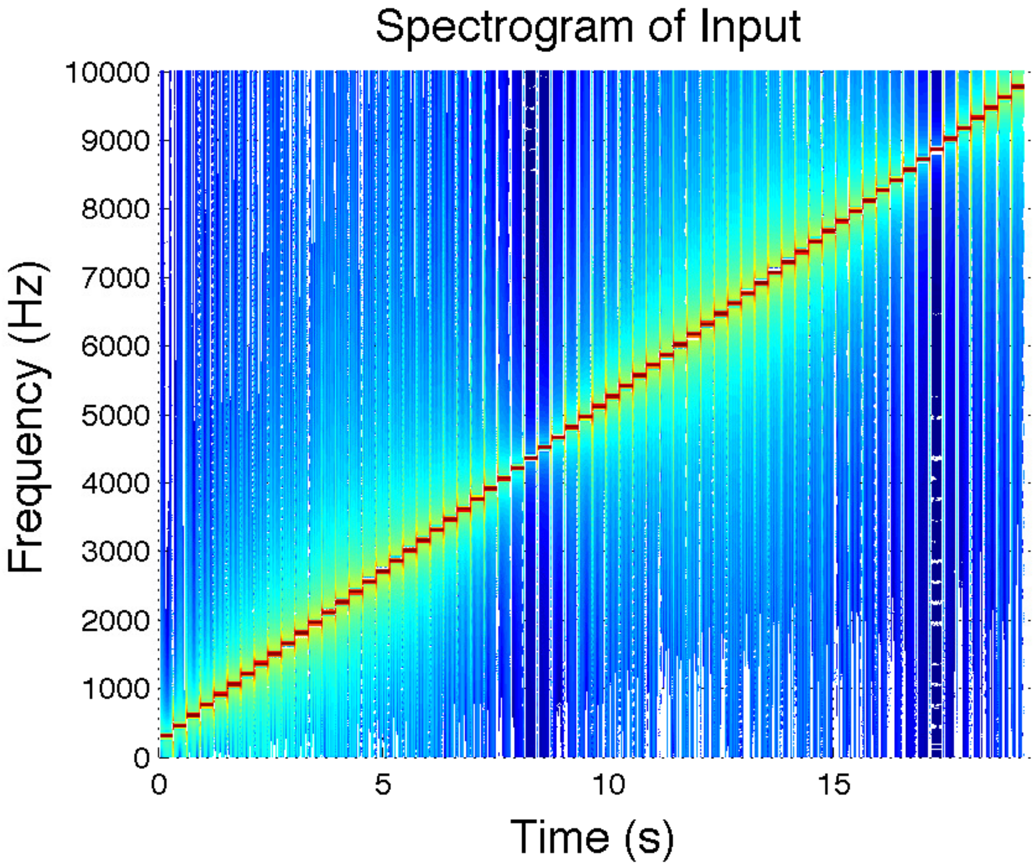


Figure 6.6: Spectrogram of the raw input signal consisting of 65 tones at 150 Hz intervals between 300 Hz and 10 kHz. Compare with Figure 6.7 which shows the spectrogram after the input signal is generated by a particular loudspeaker and recorded.

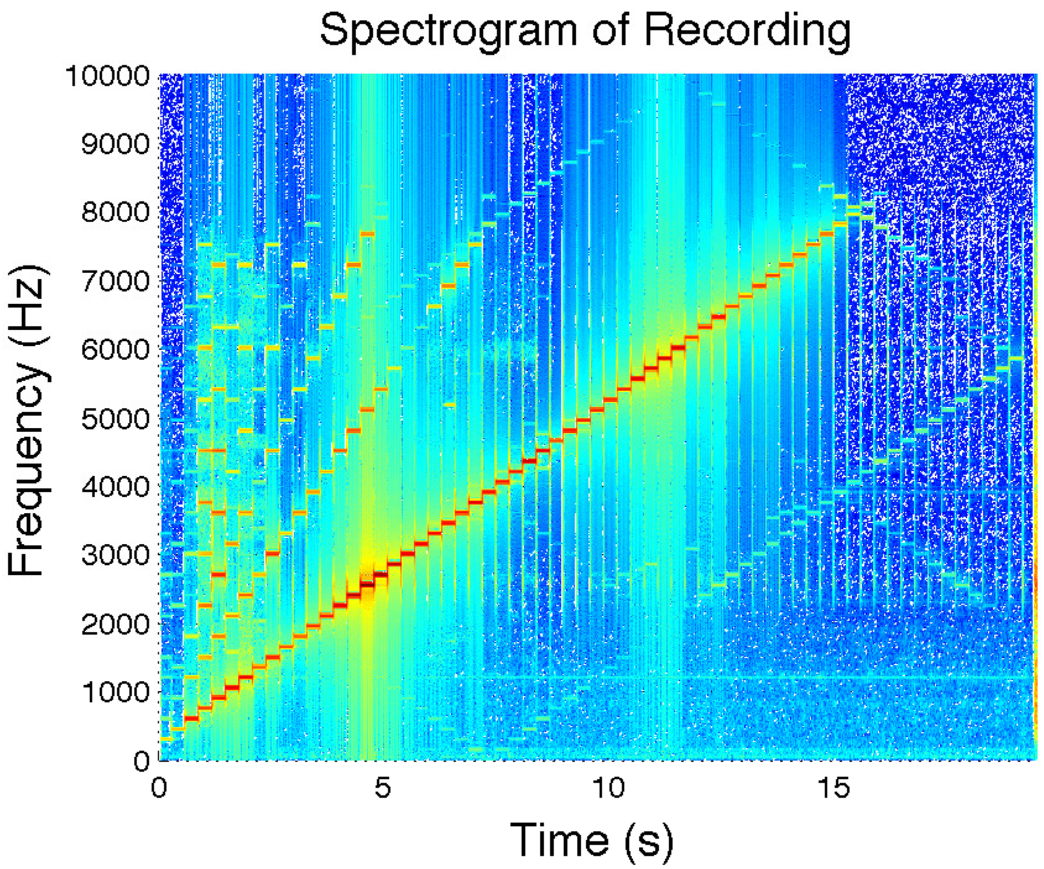


Figure 6.7: Spectrogram of recorded audio from one of the laptops used in experiment. Compare with Figure 6.6. There are a number of aliased frequencies that likely correspond to strong resonances in the speaker body or room. This particular speaker seemed incapable of generating frequencies above 8 kHz with significant magnitude.

To measure the strength with which each frequency is generated, we use the Fast Fourier Transform (FFT). In our analysis, we used the FFTW implementation of the Fast Fourier Transform [39]. FFTs take as input a zero-padded audio signal and return a measurement of the contribution of each frequency to the audio signal. For efficiency, FFTs measure the collective contributions of groups of neighboring frequencies. In our measurements, we had an FFT bin-size of 0.084114 Hz.

Our feature vector consists of frequency magnitudes around the 65 frequencies in the input signal. We performed a series of experiments to determine the optimal band of frequencies to include in the feature vector. Through experimentation, we determined that the band of +/- 5 Hz around each input frequency captures the

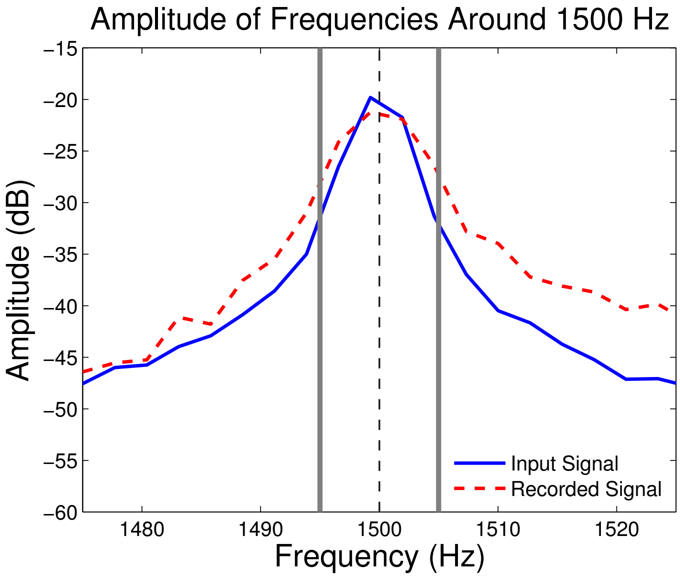


Figure 6.8: The frequency distribution of a loudspeaker's input and output signals. The loudspeaker distorted the input signal by generating stronger frequencies near the intended output frequency. Even though the input signal consists of a single 1500 Hz tone, generating a single FFT with such a large window increases the resulting frequency spread [93]. In this instance, the strength of the 1500 Hz frequency in output signal is close to that of the input. This is not always the case. Figure 6.5 shows measurements from two different machines where the magnitudes across different frequencies vary greatly. We found the majority of the distinguishing information is captured in the +/- 5 Hz band surrounding the expected frequency. For each signal, we used a Blackman window half the size of the FFT window.

majority of the distinguishing information about a loudspeaker and resulted in a reasonable length feature vector. For example, if our audio signal contained a 1500 Hz tone, our feature vector includes the measured intensities of frequency bins that fall between 1495 and 1505 Hz. See Figure 6.8, which visually denotes the band of frequencies included in the feature vector. With an FFT of size of 524288, our FFT has a bin size of 0.084114 Hz. There are up to 119 FFT bins within the 10 Hz frequency range we are interested in. The result is a feature vector made up of 65 $\textit{frequencies}$ x 119 $\frac{measurements}{frequency}$ = 7735 frequency amplitudes. Due to the particular bin size, the actual number of frequency amplitudes used was 7728. In all experiments, unless otherwise noted, our feature vector consists of 7728 frequency measurements.

6.3.3 Distinguishing Between Loudspeakers

In each of the following experiments, we generate a set of audio recordings played by a particular loudspeaker. For each audio recording, we generate a feature vector as described in the previous section. Each feature vector is labeled with the loudspeaker it came from.

We use Weka (version 3.6.3), a popular machine learning workbench with a wide selection of pre-built machine learning algorithms, to train a set of classifiers (a model) to distinguish each loudspeaker [46]. We experimented with a variety of a algorithms, including nearest neighbor, naive Bayes and multilayer perceptrons, and found the best results with Weka's implementation of the Sequential Minimal Optimization (SMO) supervised learning algorithm. SMO is an efficient method for training support vector machines. Weka accepts a training set as input and trains a set of SVMs (which we call a model) to distinguish between the input classes. Weka evaluates the trained model against a provided test set. Internally, when evaluating a model against an input example, Weka generates a list of confidences corresponding to each loudspeaker label. We developed glue-code to extract this list of confidences.

6.4 Experiments

We performed a series of experiments to measure how accurately a loudspeaker can be identified based on the distortion of a known input signal. Our experimental setup consisted of loudspeaker and microphone connected to a laptop computer. See Figure 6.9. A known signal is played through the loudspeaker and simultaneously recorded by the microphone. Audio is recorded at 44.1 kHz with a 16-bit resolution. In most experiments, out of convenience, we utilize the laptop's built-in speakers and microphone.

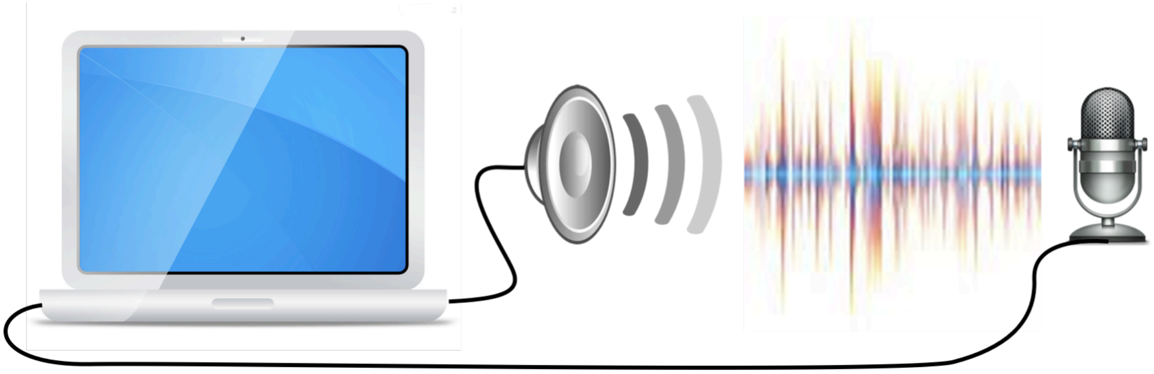


Figure 6.9: Typical Experimental Setup. A laptop computer is connected to a loudspeaker and microphone. A known audio signal is played and the generated audio is recorded by the microphone. In most cases, the loudspeaker and microphone were built-in to the laptop. Audio was recorded at 44.1 kHz with a 16-bit sample resolution.

Our first experiment focuses on distinguishing between a set of 19 homogeneous loudspeakers. A second experiment consists of measurements from a set of 107 heterogeneous machines, e.g. loudspeakers of different makes or model. We also perform experiments to determine resiliency in the presence of noise, and the ability to distinguish between loudspeakers using pink noise as the input audio signal. The results demonstrate that loudspeakers are distinguishable based on particular distortions of the same input signal, and are relatively robust to noise.

There are a number of factors that could contribute to the measured distortions. In Section 6.5 we describe a set of experiments designed to measure the impact of various factors on the observed differences. The main result from those experiments is that the loudspeaker is the largest contributor to the measured distortion (for the feature vector we used, at least), supporting accurate re-identification.

6.4.1 Identifying Homogeneous Loudspeakers

In this experiment, we are interested in measuring the ability to distinguish between otherwise identical loudspeakers. We obtained access to a set of 19 Lenovo R61 laptops that were purchased in the same lot and have remained in the same location with similar usage since their purchase. Each laptop has a pair of built-in loudspeakers of the same make and model.

To measure the characteristics of the loudspeakers built-in to each machine we placed each laptop in the center of a large classroom. Each laptop has a microphone built into the top of the screen. We took care to position the screen consistently, ensuring the microphone was in the same location relative to the loudspeaker. For each laptop, we generated a series of tones, each lasting 0.3 seconds, recording the output using the built-in microphone. On the day when the experiments were performed, we performed a set of 10 measurements on each of the machines in the morning, for a total of 190 measurements. Four hours later, we returned to the same room and re-measured the same machines in an identical fashion, resulting in an additional 190 measurements. The environmental conditions in the morning and afternoon were similar, except in the afternoon the air conditioning was running and there was noticeably more activity in the nearby hallway.

For each of the audio recordings, we generated a feature vector as described in Section 6.3.2. Each feature vector was labeled with a class equal to the ID of the machine that it came from. The feature vectors derived from the morning measurements were used as the training set. The second set of 190 measurements from the afternoon, became the test set.

We used Weka to train a classifier to distinguish between the set of loudspeakers based on their feature vectors. We were able to distinguish between the set of 19 loudspeakers with 98.42% accuracy. Only 3 out of the 190 measurements were misclassified, and all came from the same machine. This experiment validated the ability to distinguish between seemingly identical loudspeakers based solely on the distortions they introduced. If we take a majority vote from the set of 10 measurements from each machine in the test set, we achieve a 100% re-identification rate.

6.4.2 Identifying Heterogeneous Loudspeakers

To generate measurements from a large and diverse set of machines we turned to Amazon's Mechanical Turk [18]. Amazon's Mechanical Turk is an online service that matches employers and workers for micro-jobs. Employers, called "requestors," post jobs for which they are willing to pay a worker a small fee to complete. Most jobs are relatively simple, and workers are typically paid less than \$0.50 for each job they successfully complete.

We implemented a small client program in Java that workers downloaded. This program generated the same series of tones as described in Section 6.3. The duration of each tone was shortened to 0.15 seconds to encourage workers to perform multiple measurements. Each worker recorded 10 measurements of their computer's speakers using their own microphones, and uploaded the recordings to our system. For each set of 10 measurements, we paid each worker \$0.25. If a worker was willing to perform an additional 10 measurements, they were paid \$0.25 for each set of 10 measurements, up to a total of 100 measurements. The only caveat was that each set of 10 measurements must be separated by at least 10 minutes. These restrictions were enforced by the application. We also recorded demographic information about each client including their OS Version, IP, and MAC Address in order to detect if any worker attempted to circumvent the enforcement mechanism. This information was also used to improve methods of detecting measurements from the same machine.

In total, we obtained measurements from 107 distinct machines across 19 different countries. Of these 107 machines, 69 performed multiple measurements that occurred more than 10 minutes apart. For each machine, we randomly selected 10 measurements that were recorded at the same time, and included those in our training set. If measurements remained for a particular machine, we selected 10 of them at random, and included those in our test set. Using Weka, we trained our model to distinguish between each of the 107 machines.

For each of the 10 measurements from each machine in the test data, we evaluated the model. To determine a match, we chose the most likely candidate consistent with all 10 measurements. To determine the most consistent match, we averaged the list of confidences generated by the model when evaluating each measurement. The result is a best-guess accuracy of 81%. This result is the average across 10 runs, with different measurements in the training and test sets. The standard deviation was 3.8%.

Knowing the most likely candidate is not sufficient for some applications. Instead, some applications need to know the probability that the true speaker is found in the top-N most likely candidates. We re-evaluated our models using the test data, recording the frequency that the true loudspeaker was found at or above each rank. See Figure 6.10. As expected, the top choice is correct 81% of the time. The correct loudspeaker is found in the top-3 most likely candidates 86% of the time, and in the top-10 89% of the time.

6.4.3 Pink Noise

In the previous experiment, we measured the distortion of each loudspeaker when generating a *single* frequency. However, under normal operation loudspeakers generate a large number of frequencies simultaneously, e.g. when playing music or movie soundtrack. To measure the distortion of loudspeakers across a large number of frequencies simultaneously, we recorded 3 seconds of pink noise from each of the 107 loudspeakers in the previous experiment. Pink noise is similar to white noise, but differs in an important way. The power density function of pink noise decreases at a rate of -3 dB for each octave. Pink noise is often used to test the frequency response of loudspeakers because this power density drop-off is similar to how the human ear perceives sound [110].

For each measurement, we calculated the frequency response of each loudspeaker using the 3 seconds of randomly generated pink noise. The same pink noise waveform

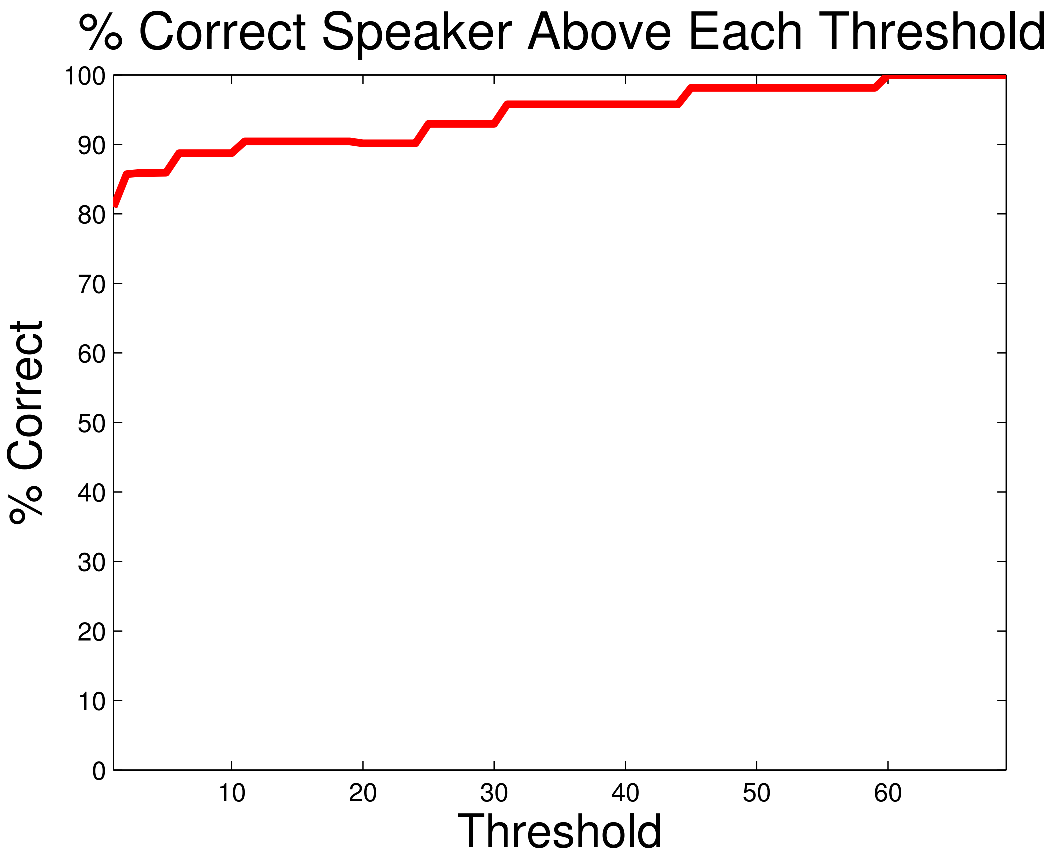


Figure 6.10: Measurements from 107 machines are used to train a classifier. 69 machines are evaluated in the test set, and the probability that the correct speaker is found at or above a particular rank is shown above. The top choice is correct 81% of the time. The true speaker is found in the top-3 most likely candidates 86% of the time, and in the top-10 89% of the time.

was played by each loudspeaker. In this case, our feature vector included every measured frequency. We trained our classifier to distinguish between each loudspeaker, and evaluated the accuracy of our classifier on measurements from 69 of these machines. Our model achieved a re-identification accuracy of 88.57%.

6.4.4 Robustness to Noise

In many applications, loudspeakers may need to be identified in the presence of background noise. We evaluated the ability to distinguish between loudspeakers even in the presence of noise using the set of 19 homogeneous loudspeakers. We simulated

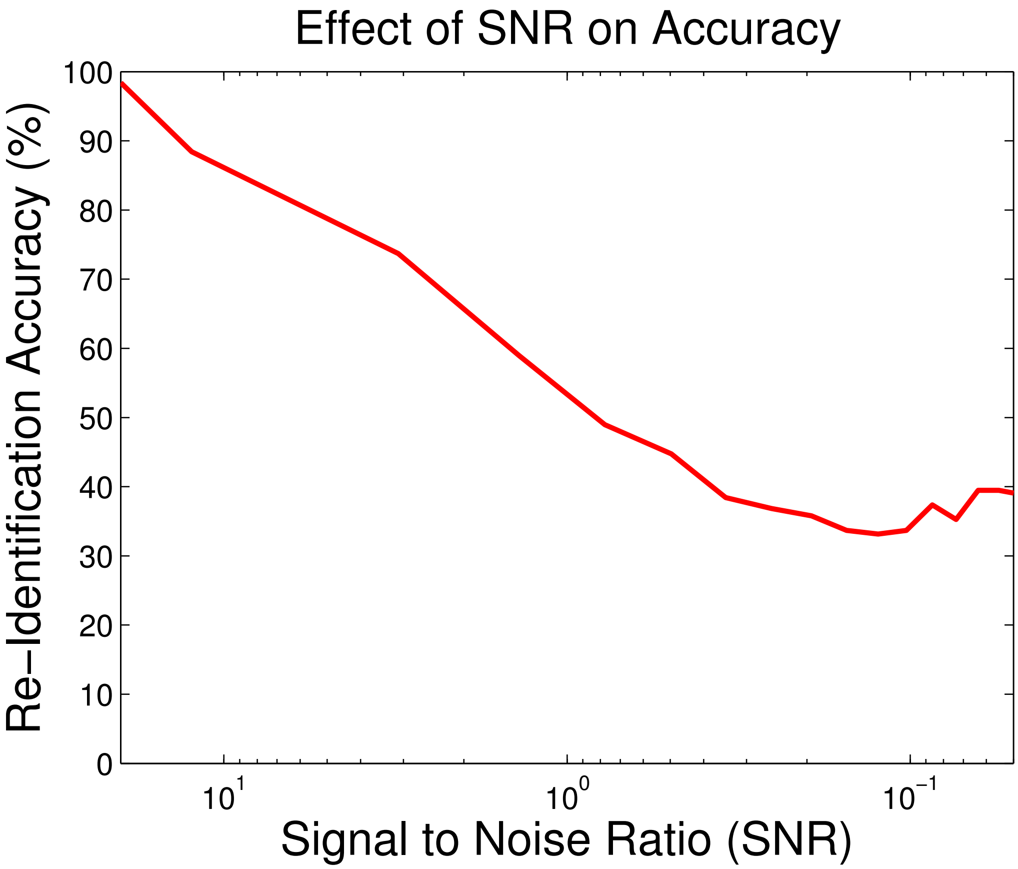


Figure 6.11: Depicts accuracy in the presence of noise. As more noise is present in the signal (decreasing SNR), re-identification accuracy decreases from 98.4 % (with clean audio), to 58.42% with SNR = 1.37, and decreases to about 40% as noise level increases.

Potential Factor	**Description**
Microphone	The microphone is responsible for measured distortion
Location	Dependent on the room acoustics, and echo.
Relative Location	The location of the speaker within a room, or position of the microphone relative to the speaker.
Time	Stability of distortion over time.
Speaker	Combinations of Components are cause of distortion
Amplifier	The amplifier to which the speaker is connected generates a distortion based on the input signal.

Table 6.1: Contributing factors to measured audio distortion

background noise by adding an increasing level of randomly generated pink noise to each measurement, measuring the signal-to-noise ratio (SNR). In this experiment both the training and test set had equivalent levels of noise added. The results, shown in Figure 6.11, demonstrate that we are still able to identify the same machine with significant accuracy, even as the SNR decreases. While these results are simulated, they provide insight into the practicality of distinguishing between loudspeakers in noisy environments.

6.5 Contributing Factors

There are a number of factors that can contribute to measurable distortion of an audio signal. See Table 6.1. The microphone, location of the speaker, location of microphone relative to speaker, time of measurement, the amplifier or the speaker itself can all contribute to measurable differences. We performed a series of experiments to determine what effect each of these factors has on the measured differences between speakers.

Loudspeaker Location: This experiment explored the impact of location on our ability to distinguish between speakers. We used a set of 22 heterogeneous machines. We placed each machine in the same location in room A (a small office), and recorded the same series of tones as described in Section 6.3. We repeated this 10 times on each machine. We then moved the laptops to room B (a conference room), generating an additional 10 measurements. We randomly selected 11 machines to be the training set, with the remaining 11 machines being in the test set. Each measurement was labeled with the room in which it was taken.

We trained our model to distinguish between rooms A and B. Our model was unable to distinguish between rooms with significant success, achieving only 55% accuracy, only 5% better than random guessing. The model's inability to distinguish between rooms is likely a result of the way we generate our feature vector. The feature vector consists of the intensities of frequencies in the recorded audio which are unlikely to capture room specific characteristics, such as echo. We concluded that the room in which a speaker is located plays only a negligible role in the ability to distinguish between loudspeakers based on our method of selecting features.

There are a number of existing methods that can distinguish between locations based on other audio-based methods, e.g. using delay based techniques or background audio noise to generate feature vectors [60, 95]. We chose our feature vector in a way that minimizes the impact of delay and background noise by focusing on frequencies nearest to those we expect to be present. While the location in which a recording is made can have an impact on the recorded audio signal we saw no measurable impact in our experiments.[1]

Relative Location: Our inability to accurately distinguish between locations with dramatically different acoustic responses suggests that our ability to distinguish between measurements taken at relatively similar locations will be quite low. We did not investigate this further.

Stability Across Locations: In light of the previous experiment demonstrating our inability to distinguish between locations, we wanted to measure the stability of loudspeaker distortion across locations. In this experiment, we took 10 measurements in Location A, but evaluated our model based on 10 measurements taken in Location B. We used a set of 8 heterogeneous devices.

We trained our model on the measurements taken at Location A, and evaluated our model based on the measurements taken at Location B. The model was able to distinguish the correct device 94.4% of the time. This demonstrates that a speaker

[1] One of the strongest location based effects is resonance. The resonant frequencies (also called room modes) of a room depend on its size, shape and material properties. A room's mode can be approximated via $f = \frac{v}{2d}$, where v = 1130 $\frac{feet}{second}$, the velocity of sound in air, and d is the width, height or depth of the room [77]. For reasonable values of d, the primary room mode is well below 300 Hz [30,97]. In our experiments, we do not measure the distortion of frequencies below 300 Hz and while multiples of the primary room mode may have an effect, their impact will be attenuated [97].

is distinguishable even if training and testing occur at different locations. This result reinforces the notion that our method of feature vector generation captures properties of the speaker, not the room in which the measurements are taken.

Microphone: In this experiment, we selected two otherwise identical, 2008 Apple MacBook laptops. We placed each machine in the same location in the same room. To generate training data, we connected a pair of Altec Lansing BXR 1220 speakers to each machine. We collected a set of 10 measurements from each machine. Each machine's built-in microphone recorded the audio. Each measurement was labeled 'Mic1' or 'Mic2' corresponding to the each machine's microphone. To generate test data, we collected an additional set of 10 measurements from each machine using each device's built-in speakers with each measurement labeled with the machine that it came from. We trained our classifier using the training data and evaluated the ability to distinguish between microphones using the test data. We were unable to distinguish between each machine's built-in microphone above 50% accuracy.

Amplifier: In virtually all settings we are concerned with, the amplifier is built into the machine to which loudspeakers are connected. This provides a 1-1 mapping between the generated audio of the loudspeaker and amplifier built-in to the machine it is connected to. We did not attempt to identify whether the amplifier contributed to the perceived distortion.

Speaker Components: We did not explore the impact that each sub-component of a speaker has on the measured distortion. We treat the speaker as a single unit. While some audio enthusiasts customize their speakers, perhaps using different diaphragms or drivers, most loudspeaker owners do not.

6.6 Authenticating Loudspeakers

Some applications rely on the ability to 'fingerprint' a particular loudspeaker. This is different than distinguishing a loudspeaker from among a known set. In this case, a 'fingerprint' of a loudspeaker is generated at time t_1. When a loudspeaker is presented for identification, at time t_2, its distortion is re-measured and compared against a claimed identity, or a database of loudspeaker 'fingerprints' to determine the true identity. Device authentication is one application where the ability to 'fingerprint' loudspeakers is necessary.

Device authentication typically begins with a a registration stage, where a particular property of the device is measured. At some point in the future, the device must be identified, passing or failing the verification stage. We describe our loudspeaker registration and verification processes and evaluate the ability to identify particular loudspeakers.

6.6.1 Registration

To generate a 'fingerprint' for a loudspeaker, we measure the characteristics of the frequency response. A loudspeaker's frequency response is measured 10 times, just as in Section 6.3.2. We take the average of these frequency responses. This becomes the fingerprint for a loudspeaker. Depending on the application, this fingerprint can be either stored in a database or digitally signed by the registration entity.

6.6.2 Verification

When a loudspeaker is presented for verification, its frequency response is compared to that of the claimed identity. If no particular identity claim is made, then it is compared against a database of 'fingerprints', identifying the one with the closest match, above a certain threshold. We compare fingerprints based on the Pearson Correlation coefficient, $Corr(X,Y) = \frac{\sum(X_i-\overline{X})*(Y_i-\overline{Y})}{\sqrt{\sum(Y_i-\overline{Y})^2*\sum(X_i-\overline{X})^2}}$, between the loudspeaker's true fingerprint and the claimed fingerprint [38] . Prior to correlation, each of fingerprints is normalized to the interval [0, 1.0] . The loudspeaker matches the claimed identity if the correlation is above a certain threshold, otherwise the claim is rejected.

6.6.3 Evaluation

Using the same set of 107 loudspeaker measurements used in Section 6.4.2, we evaluate loudspeaker fingerprints as a method to verify a loudspeaker's identity claim. In this experiment, we take the Pearson Correlation between the loudspeaker's fingerprint, at t_2, and its previously measured fingerprint at t_1. We accept the loudspeaker's identity claim if the correlation falls above a certain threshold. The results, shown in Figure 6.12, show that with a correlation threshold of 0.844, we will accept a machine's correctly asserted identity with 84.5% accuracy.

Many applications are concerned not only with the True Positive Rate (accepting the correct loudspeaker), but with minimizing the intersection of False Positive and False Negative Rates. The False Positive Rate is the probability that a impostor loudspeaker will be accepted as the claimed loudspeaker. Conversely, the False Negative Rate is the probability that the claimed loudspeaker will fail to be accepted. Figure 6.12 shows the True Positive Rate as a function of correlation. With a correlation threshold of 0.75, we achieve an acceptance rate of 89.9%. Figure 6.13 describes the False Positive and False Negative Rates as a function of correlation threshold. We achieve an equal-error-rate of 15.5% at a correlation threshold of 0.78.

6.7 Authentication

Consider the situation where Bob wants to authenticate himself to Alice. As part of a multi-factor authentication strategy, Alice can authenticate Bob's loudspeakers.

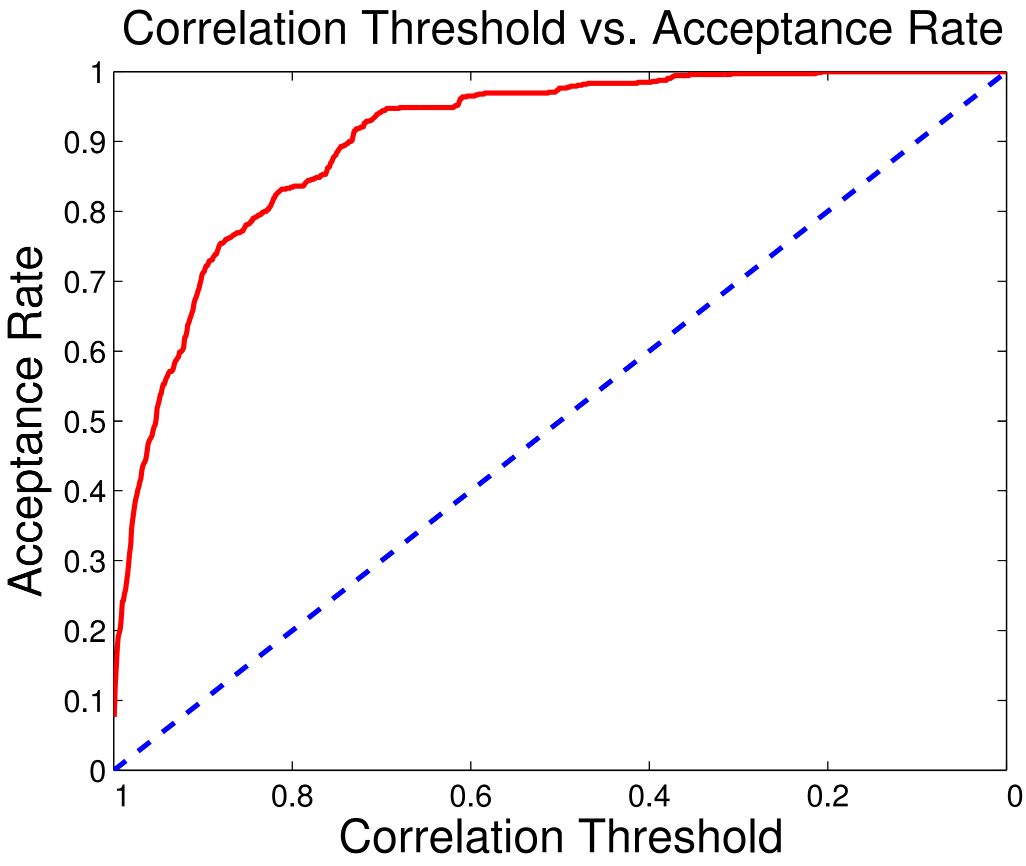


Figure 6.12: For each of the machines in the 107 machines gathered via Mechanical Turk we calculated the fingerprint for each device based on the average response from 10 measurements. We compared this average response with any subsequent measurements from the same by calculating the Pearson Correlation. As the correlation threshold decrease from 1.0 to 0, the percent of accepted machines increases. The dashed line is what we would expect to see for correlations between random signals.

We'll assume that Alice and Bob met in the past, and Alice measured the distortion properties of Bob's loudspeakers.

When Alice receives an authentication request from someone claiming to be Bob, she will try to verify that it originated from Bob's computer. To do this, Alice generates a challenge—a unique audio signal—for each login attempt. For example, Alice might choose an audio signal that is drawn randomly from a pink noise distribution. Bob's computer will play and simultaneously record the audio signal and submit the recording to Alice. Using the techniques described in Section 6.3.3, Alice will accept or reject the login attempt.

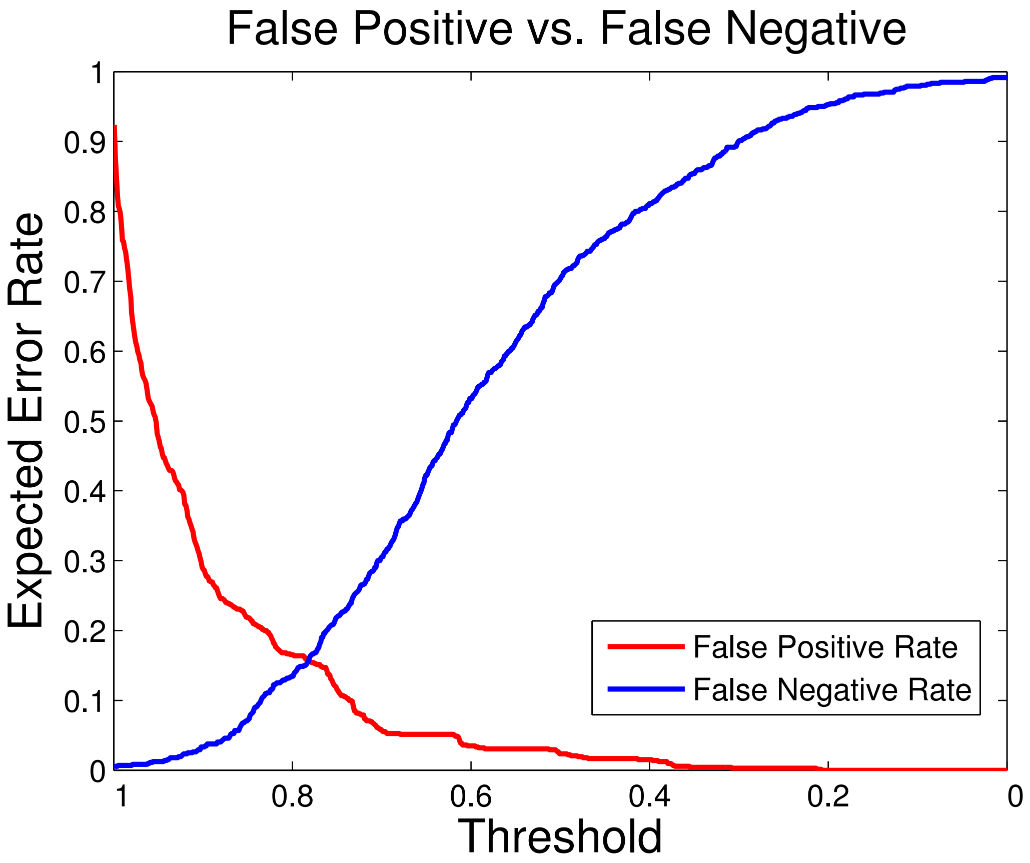


Figure 6.13: Depicts variations in the False Positive and False Negative rates as the threshold varies. The False Positive rate measures the probability that an impostor loudspeaker is mistakenly accepted for the true loudspeaker. The False Negative rate measures the probability that the true loudspeaker will be incorrectly identified as an impostor. They intersect at 15.5% where the correlation is 0.78.

Given the level of success in the previous sections, this technique can authenticate Bob to a certain degree of confidence. Depending on the application, this may not be enough. Some applications may want to prevent an eavesdropper, Eve, from learning anything meaningful about Bob's loudspeakers. Under this threat model, Eve is able to hear the sounds that Bob's loudspeakers make. This the auditory equivalent of shoulder surfing.

If Eve is able to listen to Bob authenticate to Alice multiple times, Eve may (or may not) be able to learn the distortions of Bob's loudspeakers and impersonate Bob. In this threat scenario, Eve is unable to simultaneously know the generated output and the specific input that generated it. We further assume that Bob and Alice share a cryptographic key and utilize a protocol, like SSL, that can detect a man-in-the-middle attack where Eve might attempt to relay responses between Alice and Bob [31].

We define $D(S)$ as the distortion on an input signal S induced by Bob's loudspeaker. $D^{-1}(S)$ is the inverse of the distortion function.

The security properties of a loudspeaker fingerprint depend on the ability of Eve to mimic the characteristic distortions of Bob's loudspeakers. We are still evaluating the ability of Eve to mimic a specific loudspeaker. In light of that, even if Eve is able to mimic Bob's speakers, we believe the authentication scenario may remain viable.

If Alice knows D^{-1} exactly, then she can authenticate Bob in zero-knowledge. Eve chooses a nonsecret distribution C over sounds. To authenticate Bob, Alice uniformly samples a sound c from C, and sends $D^{-1}(c)$ as the challenge to Bob. Bob's speakers distort the sound to $D(D^{-1}(c)) = c$. He sends the result back to Alice, and she authenticates the result as consistent with Bob's distortion function. Eve gains zero knowledge, because she learns only the audible sound c which is drawn uniformly from the non-secret distribution C. For example, if C is the pink noise distribution, then Eve will hear pink noise which does not convey anything about Bob's speakers.

In practice, Alice will not know D^{-1} exactly but will only know an approximation $D^{*-1}(S) = D^{-1}(S) + \epsilon(S)$ where ϵ is ideally close to zero. In this case Eve will see a distribution that differs from C. However, if the statistical difference between C and the distribution Eve sees is small, then Eve will not have a substantial advantage over random guessing. For example, if Eve has advantage α (compared to random guessing) in distinguishing the observed distribution from C, then Eve will have at most advantage α in impersonating Bob.

6.8 Applications

The ability to identify a particular speaker based on measurable distortions makes a number of applications possible. This technique also has implications for user privacy. Two interesting applications are user tracking and authentication. The major distinction between tracking and authentication is whether the user consented to being identified.

6.8.1 Authentication

This technique works on on any existing device that has a built-in speaker or microphone. As described in the previous section, this technique can be used as a means of authenticating a device. This can be an alternative form of two-factor authentication. In traditional two-factor authentication, the user must present something they have, usually a physical token, e.g. ATM Card, RFID, Smart Card, One-Time-Pad, and something they know, e.g. their password or pin number. If we can authenticate the device from which they are logging in, then the device could substitute for, or be used in addition to, the 'thing they have'.

This may be important in applications, such as online banking and access control. Content providers may wish to limit the number of devices that a user can use to access content. For example, video-game manufacturers could use this technique to restrict access to games, except from pre-registered devices. This would prevent users from sharing their games, as the loudspeakers used at a second location will, with high likelihood, be identifiably different than those of the owner. Alternatively, a user may elect to restrict the particular set of devices that are allowed to access his or her account. Anytime a user logs in, e.g. to their bank, this may require the device to identify itself using a simple challenge response protocol.

6.8.2 User Tracking

Most users do not change their speakers very often. Users with laptops or smartphones are highly unlikely to change their speakers as doing so may void their warranty. In most instances, identifying a particular set of speakers also identifies the machine they are connected to, and in turn the identity of the user. This, combined with the ability of Java and Flash applications to access to a computer's microphone, even when running inside the browser, allows loudspeakers to act as a sticky physical cookie. A user could be re-identified even if they clear their cookies. While the user's permission is required the first time the microphone is accessed clever integration of the microphone into the particular use case mitigates this hurdle [4].

For example, many online games play music during the course of play, e.g. Angry Birds. In some of these games, the game can attempt to access the built in microphone on the device – perhaps by building an aspect that relies on microphone input. A malevolent developer could try to link otherwise distinct sessions based on similarities between a user's physical device features.

6.9 Future Work

There are a number of potential avenues for future work. The most pressing is duplicating this work on a variety of devices, e.g. headphones and cell phones. We are also working to collect measurements from a larger set of speakers, especially those of the same make and model, to strengthen the experimental results shown here.

An additional application that should be explored is caller device identification. It may be possible to use this technique to identify whether a caller is using the same device as the last time they called. By measuring the 'echo' in the call, the device on the other end may be fingerprinted. This method may be complicated by the increasing use of noise cancellation and the compression of audio transmitted via networks.

6.10 Discussion

This work challenges the assumption that seemingly identical loudspeakers are indistinguishable from one another. We identified and characterized the distortion introduced by ordinary, unmodified, loudspeakers. We then developed techniques to distinguish between loudspeakers by measuring the unique way that each loudspeaker distorts a known audio signal.

To measure the ability to distinguish between loudspeaker, we performed a series of experiments on a set of 19 otherwise identical loudspeakers. We were able to distinguish between them with over 98% accuracy. We also explored the ability to distinguish between measurements from much larger heterogeneous set of loudspeakers gathered sing Amazon's Mechanical Turk. We measured 107 loudspeakers, distinguishing between them with 81% accuracy.

The ability to distinguish between loudspeakers enables a number of applications, such as a new way to authenticate devices, and has potential implications for user privacy.

Chapter 7

Conclusion

The landscape of what is measurable has fundamentally changed. In the past decade, the number of sensors people carry with them on a daily basis has dramatically increased. The Google Nexus S cell phone, released in 2011, contains over 11 different kinds of sensors, not including the Wi-Fi and CDMA/GSM receivers. These additional sensors include an accelerometer, barometer, Bluetooth^{TM}, camera (both still and video), compass, GPS, gyroscope, loudspeaker, Near-Field Communications (NFC), two microphones and capacitive touch screen. The evolution of the cell phone from a device originally designed to make voice calls into a multi-purpose, multi-input, multi-sensor device is one tangible impact of the increasing availability of highly accurate, cheap sensors. In fact, many other sensor devices, e.g. traffic cameras, cars, and smart buildings are experiencing similar growth in the number and variety of cheap sensors. This explosion of accurate sensors in the hands of millions of people creates an increasing potential to easily measure and eventually trace commonplace items.

To identify and distinguish between items, new methods are needed that can utilize cheap sensors to measure an item's unique properties. This thesis examined three different commonplace artifacts: blank sheets of paper, bubble-forms, and loudspeakers. In each case, we utilized cheap sensors, signal processing and machine learning to identify unique properties in these otherwise seemingly identical objects. We repeatedly challenged the false assumptions regarding the indistinguishability of each artifact. We showed how violating this common, false assumption can have an immediate, negative impact on applications that explicitly or implicitly rely on an item's indistinguishability. On the other hand, the ability to uniquely identify items enables many exciting new applications.

Through the process of developing methods to distinguish between these three kinds artifacts, a common methodology emerged. We call this methodology the General Framework. Introduced in Chapter 2 and repeatedly applied throughout this thesis, the success of the General Framework suggests that there are other items, which we currently think of as identical, that are distinguishable using cheap sensors.

We began in Chapter 3 with blank sheets of paper. In many applications, blank sheets of paper are treated as if they are identical to one another. This false treatment can lead to unintended consequences. We introduced a new method to uniquely identify blank sheets of paper by measuring their inherently unique surface texture using commodity desktop scanners. Our method improves on previous approaches in a number of significant aspects. First, it relies on physical property which is difficult to forge, a document's surface texture. Second, it does not require modification of the target document in any way. This permits the imperceptible tracking of documents. Finally, our technique utilizes inexpensive, commodity equipment, and is robust to harsh treatment. Even after soaking, crumpling, scribbling and printing, documents were re-identifiable with 100% accuracy. The ability to identify documents, even under harsh treatment, enables a number of applications, e.g. counterfeit currency detection, and has important implications for paper-based elections.

The implications of paper fingerprinting for paper-based elections are both positive and negative. The negative implications center on violating the secret ballot —

permitting voter coercion. To mitigate this risk, we suggest multiple procedural safeguards, e.g. limiting access to ballot stock. Paper fingerprinting also leads to positive applications to voting, providing a new method of detecting ballot box stuffing and significant economic savings when conducting post-election audits.

In Chapter 5, we re-evaluated a common assumption about bubble-forms: that they do not reveal the respondent's identity. We showed that this assumption is false and demonstrated that individuals tend to mark bubbles in distinctive ways, unintentionally conveying their identity by simply marking a bubble. Using a corpus of surveys taken from 92 individuals, we were able to identify the correct respondent over 51% of the time, and we locate the correct respondent in the top 10 most likely candidates 92% of the time. This result has important implications for anonymous surveys and the publication of completed ballots after an election. We described a number of mitigation techniques and procedural changes to allay the risk of accidentally revealing identifying information through bubble-forms.

Our final artifact, loudspeakers, are increasingly ubiquitous devices. While it is commonly believed that multiple instances of 'identical' loudspeakers generate the same output when given identical input, we exposed this false assumption. We demonstrated that individual loudspeakers, even those of the same make and model, induce a unique distortion on the generated sound. This distortion is easily measurable using cheap microphones an some signal processing. We developed methods to identify loudspeakers based on this distortion with high accuracy. This technique also enables a new method of device authentication.

The features of the artifacts that ultimately led to a method of identifying individual items have a common thread – they are all inherent to the item under study. The concept of measuring an inherent feature, or defect, combined with cheap sensors will inevitably lead to more items becoming traceable. The unavoidable result is that every physical item can be tracked – the question is only when, and at what cost. As sensors improve, the properties used to identify each item will become increasingly imperceptible to casual inspection, eventually leaving no indication that an item is being traced.

The long term implications of ubiquitous, imperceptible traceability will not be a net positive effect for society at large. The number and availability of accurate sensors will increasingly allow for the continuous tracking of individuals. A number of systems currently provide the ability to track individuals based on the radio transmitters in their cell phones [100]. The most common system, known in the law enforcement community by its generic name, “Stingray”, tracks cellular phones by mimicking a cell phone tower, coercing any phones in the area (hopefully including the intended target’s), to mistakenly connect to it. These fake ‘towers’ are able to measure signal strength and phone identifier information. By combining relative signal strength measurements from a number of different ‘towers’, the Stingray system can pinpoint the physical location of phones in its target area, even those devices located within private residences. While this powerful law enforcement tool may ostensibly be used to track the bad guys, the potential for misapplication is one potentially serious negative implication. The Supreme Court recently ruled that tracking individuals by attaching GPS transmitters to their cars requires a warrant, i.e. it is trespassing to attach the transmitter to the vehicle [94]. However, the Court has not determined whether a warrant is needed to track individuals via GPS transmitters already in place, e.g. like the kind commonly found in cell phones or built in to newer cars [63].

The implications for personal privacy by government actors may be secondary to invasions by the commercial sector. With an ever larger number of commercially available devices that include cheap, accurate sensors, we will see an increasing attempt by companies to collect and maintain information about the usage of these devices. The recent attempt by OnStar, a safety and connectivity service for automobiles, provides a perfect example of the potential privacy risks. In 2011, OnStar attempted to rewrite its terms and conditions to allow the sale of customer's 'anonymized' GPS data to a third-party [87]. Interestingly, even if a customer cancelled their service, their vehicle's location data would continue to be collected and sold if the customer did not explicitly opt-out of the tracking. After receiving letters from several senators, OnStar cancelled the proposed changes.

Even popular technology companies, like Apple Inc., are not immune from the potential privacy risks presented by ubiquitous sensors. In early 2011, the public learned that each iPhone maintained a record of the device's estimated GPS locations. These location records were transferred to the user's computer during each sync [14]. Once aware, the public responded negatively and senators wrote letters asking Apple for more information about the scope and nature of the location tracking. The bad press caused Apple to release a software update correcting the issue shortly thereafter.

Access to location tracking technology is no longer limited to well-funded institutions like the U.S. government and Fortune 500 companies. For under 300 dollars, any member of the public can purchase a cigarette-size GPS transmitter that can be surreptitiously attached to any vehicle [36]. This enables parents to invisibly track their children, husbands to track their wives, and private investigators to track whomever they are hired to follow. In the past it took constant visual contact to track someone, effectively limiting the scale and duration of surveillance. Today, all it takes is 10 minutes to attach a transmitter to any car and the results in up to the second location information. This should give people pause. The societal trend permitting the unfettered ability to constantly monitor every individual is likely to have deleterious effects.

The development of new capabilities to track items, and by proxy people, requires a re-evaluation of the limits of government and consumer privacy protections in order to stymie the potentially negative implications. As a society, we must have an honest debate about the sacrifices we are willing to make in the name of national security and personal safety. Increasingly, these sacrifices seem to come at the cost of personal privacy. Unfortunately, the national conversation implicitly relies on the false belief that safety and privacy are a zero-sum game; one can not have more of one without sacrificing the other. This is a dangerous assumption to make, and one I hope can be swiftly discarded. With respect to consumer privacy, consumers must demand transparency in the type of data being collected and what companies can do with it. Ideally, U.S. citizens would demand a consumer protection law similar to the European Union's Data Protection Directive [80]. While not perfect, the Data Protection Directive defines a strong right to personal data privacy and strict penalties for companies that violate it. Unfortunately, given the current domestic political landscape, neither of these are likely to happen.

The increasing prevalence and accuracy of sensors, combined with the ability to invisibly track a wide variety of items and, by proxy, people will have important societal implications going forward. The ability of private companies or government agencies to track people high accuracy or know the results of the secret ballot will have important chilling effects on the expression of personal freedoms. The future depicted in Orwell's *1984* [79], with the ubiquitous tele-screens monitoring one's every movement, may not be fantastical fiction.

Bibliography

[1] Humboldt County Election Transparency Project. http://www.humetp.org/.

[2] New Jersey Statutes Annotated 19:4-13 (1976). http://lis.njleg.state.nj.us/cgi-bin/om_isapi.dll.

[3] Speaker diagram (side view). http://www.soundonmind.com/files/ speaker diagram%201%20-%20 lables%20(flat).png, October 2011.

[4] Adobe Inc. Actionscript® 3.0 reference for the Adobe® Flash® platform. http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/.

[5] Alameda County, California. Alameda County Voting System Demonstration. http://www.acgov/org/rov/votingsystemdemo.htm.

[6] Lyn S Amine and Peter Magnusson. Cost-benefit models of stakeholders in the global counterfeiting industry and marketing response strategies. *Multinational Business Review*, 15(2):1–23, 2007.

[7] Jenny Anderson and Winnie Hu. 20 Students Now Accused in L.I. Case on Cheating. *The New York Times*, 2011. http://www.nytimes.com/2011/11/23/education/more-students-charged-in-long-island-sat-cheating-case.html.

[8] A. W. Appel. Effective audit policy for voter-verified paper ballots in New Jersey, February 2007. http://www.cs.princeton.edu/~appel/papers/appel-nj-audits.pdf.

[9] Javed A. Aslam, Raluca A. Popa, and Ronald L. Rivest. On auditing elections when precincts have different sizes. In *Proc. 2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '08)*.

[10] Javed A. Aslam, Raluca A. Popa, and Ronald L. Rivest. On estimating the size and confidence of a statistical audit. In *Proc. 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '07)*.

[11] Crispian Balmer and Ken Wills. Beijing games hit by internet ticket scam. *Reuters*, Aug. 4, 2008. http://www.reuters.com/article/2008/08/04/us-olympics-tickets-scam-idUSPEK25562820080804.

[12] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo images database. `http://www.wisdom.weizmann.ac.il/~vision/photostereo/data.html`, 2007.

[13] John Bethencourt, Dan Boneh, and Brent Waters. Cryptographic methods for storing ballots on a voting machine. In *In Proceedings of the 14th Network and Distributed System Security Symposium*, pages 209–222, 2007.

[14] Nick Bilton. Tracking File Found in iPhones. *The New York Times*, 2011. `http://www.nytimes.com/2011/04/21/business/21data.html`.

[15] Justin Brickell and Vitaly Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *Proc of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2008.

[16] Benedict Brown, Corey Toler-Franklin, Diego Nehab, Michael Burns, Andreas Vlachopoulos, Christos Doumas, David Dobkin, Szymon Rusinkiewicz, and Tim Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Trans. Graphics (Proc. SIGGRAPH 2008)*, page 84 (9 pp.), August 2008.

[17] James D. R. Buchanan, Russell P. Cowburn, Ana-Vanessa Jausovec, Dorothée Petit, Peter Seem, Gang Xiong, Del Atkinson, Kate Fenton, Dan A. Allwood, and Matthew T. Bryan. Forgery: 'fingerprinting' documents and packaging. *Nature*, 436:475, 2005.

[18] Michael D. Buhrmester, Tracy Kwang, and Samuel D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, January 2011.

[19] John C. and Tipper. A straightforward iterative algorithm for the planar voronoi diagram. *Information Processing Letters*, 34(3):155 – 160, 1990.

[20] J. A. Calandrino, J. A. Halderman, and E. W. Felten. In defense of pseudorandom sample selection. In *Proc. 2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '08)*.

[21] J. A. Calandrino, J. A. Halderman, and E. W. Felten. Machine-assisted election auditing. In *Proc. 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '07)*.

[22] Joseph A. Calandrino, William Clarkson, and Edward W. Felten. Some consequences of paper fingerprinting for elections. In David Jefferson, Joseph Lorenzo Hall, and Tal Moran, editors, *Proceedings of EVT/WOTE 2009*. USENIX/ACCURATE/IAVoSS, August 2009.

[23] Joseph A. Calandrino, William Clarkson, and Edward W. Felten. Bubble trouble: Off-line de-anonymization of bubble forms. In *Proc. 20th USENIX Security Symposium*, August 2011.

[24] Richard Carback. How secret is your secret ballot? part 1 of 3: Pattern voting. `https://scantegrity.org/blog/2008/06/16/how-secret-is-your-secret-ballot-part-1-of-3-pattern-voting/`, June 16 2008.

[25] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, and A. T. Sherman. Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *Proc. 2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '08).*

[26] William Clarkson, Tim Weyrich, Adam Finkelstein, Nadia Heninger, J. Alex Halderman, and Edward W. Felten. Fingerprinting blank paper using commodity scanners. In *Proc of IEEE Symposium on Security and Privacy*, May 2009.

[27] College Board. 2010 college-bound seniors results underscore importance of academic rigor. `http://www.collegeboard.com/press/releases/213182.html`.

[28] A. Cordero, D. Wagner, and D. Dill. The role of dice in election audits—extended abstract. In *IAVoSS Workshop on Trustworthy Elections 2006.*

[29] Russell P. Cowburn and James David Ralph Buchanan. Verification of authenticity. US patent application 2007/0028093, July 2006.

[30] Trevor J. Cox, Peter D'Antonio, and Mark R. Avis. Room sizing and optimization at low frequencies. *J. Audio Eng. Soc*, 52:640–651, 2004. `http://www.aes.org/e-lib/browse.cfm?elib=13011`.

[31] T. Dierks and E. Rescorla. RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2. Technical report, August 2008.

[32] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–137, 2008.

[33] Stark Draper, Prakash Ishwar, David Molnar, Vinod Prabhakaran, Daniel Schonberg, and David Wagner. An Analysis of Empirical PMF Based Tests For Least Significant Bit Image Steganography. In *Proceedings of the Information Hiding Workshop*, 2005.

[34] Stuart Dredge. Shazam raises 32 Million Dollar funding round for TV tagging expansion. *The Guardian*, 2011. `http://www.guardian.co.uk/technology/appsblog/2011/jun/22/shazam-funding-tv-tagging`.

[35] Cynthia Dwork. Differential privacy. In *Proc of the 33rd International Colloquium on Automata, Language and Programming*, July 2006.

[36] Erik Eckholm. Private snoops find gps trail legal to follow. http://www.nytimes.com/2012/01/29/us/gps-devices-are-being-used-to-track-cars-and-errant-spouses.html, January 2012.

[37] Miro Enev, Sidhant Gupta, Tadayoshi Kohno, and Shwetak N. Patel. Televisions, Video Privacy, and Powerline Electromagnetic Interference. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011)*, 2011.

[38] David Freeman, Robert Pisani, and Roger Purves. *Statistics*. W. W. Norton & Company, 4th edition, 2007.

[39] Matteo Frigo, Steven, and G. Johnson. The design and implementation of FFTW3. In *Proceedings of the IEEE*, pages 216–231, 2005.

[40] Trip Gabriel. Cheaters find an adversary in technology. *New York Times*, December 27 2010. http://www.nytimes.com/2010/12/28/education/28cheat.html.

[41] Trip Gabriel. Under pressure, teachers tamper with tests. *New York Times*, June 10 2010. http://www.nytimes.com/2010/06/11/education/11cheat.html.

[42] Calvin H. Goddard. Who did the shooting? *Popular Science*, 111:21–22, November 1927.

[43] Thadeus Greenson. Humboldt county transparency project now online. *Times-Standard*, 2008. http://www.times-standard.com/ci_9919694.

[44] Thadeus Greenson. Software glitch yields inaccurate election results. *Times-Standard*, December 5 2008. http://www.times-standard.com/ci_11145349.

[45] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. Electrisense: Single-point sensing using emi for electrical event detection and classification in the home. In *UbiComp*, September 2010.

[46] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[47] Ordway Hilton. The complexities of identifying the modern typewriter. In *Journal of Forensic Sciences*, 1972.

[48] Keith Inman and Norah Rudin. *Principles & Practice of Criminalistics The Profession of Forensic Science*. CRC Press, 1 edition, August 2000.

[49] J. Edward Jackson. *A User's Guide to Principal Component Analysis.* Wiley-Interscience, 2003.

[50] A. Jain, L. Hong, and S. Pankanti. Biometric Identification. *Communications of the ACM*, 43(2):91–98, February 2000.

[51] K. C. Johnson. Election certification by statistical audit of voter-verified paper ballots, October 2004. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=640943`.

[52] I. T. Jolliffe. *Principal Component Analysis.* Springer, second edition, October 2002.

[53] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proc. 6th ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[54] Kenichi Kanatani. Optimal homography computation with a reliability measure. In *IAPR Workshop on Machine Vision Applications*, 1998.

[55] Arthur M. Keller and David Mertz. Privacy issues in an electronic voting machine. In *In Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES*, pages 33–34. ACM Press, 2004.

[56] Nitin Khanna, Aravind K. Mikkilineni, George T. C. Chiu, Jan P. Allebach, and Edward J. Delp. Scanner identification using sensor pattern noise. In Edward J. Delp III and Ping Wah Wong, editors, *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, page 65051K. SPIE, February 2007.

[57] Nitin Khanna, Aravind K. Mikkilineni, and Edward J. Delp. Scanner identification using feature-based processing and analysis. *Trans. Info. For. Sec.*, 4:123–139, March 2009.

[58] Steven D. Levitt and Stephen J. Dubner. *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything.* HarperCollins, 2006.

[59] Los Angeles County Registrar-Recorder / County Clerk. InkaVote Plus Manual. `http://www.lavote.net/voter/pollworker/PDFS/INKAVOTE_PLUS_HANDBOOK.pdf`, 2011.

[60] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 165–178, New York, NY, USA, June 2009. ACM.

[61] J Luka, J Fridrich, and M Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.

[62] D.J.C. MacKay and R.M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 33(6):457–458, March 1997.

[63] Mike Masnick. Fourth Amendment Lives? Supreme Court Says GPS Monitoring Is A Search That May Require Warrant. `http://www.techdirt.com/articles/20120123/11261317515`, January 2012.

[64] Chuck McCutcheon. Absentee voting fosters trickery, trend's foes say. *Times Picayune*, October 2006. `http://www.nola.com/news/t-p/frontpage/index.ssf?/base/news-6/116168644619990.xml&coll=1`.

[65] William K. McFadden. Loudspeakers primer. `http://www.rdrop.com/~billmc/spkr.txt`, January 1999.

[66] D. McNicol. *A Primer on Signal Detection Theory*. Lawrence Erlbaum Assoc., 2004.

[67] Frank D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of 35th SIGMOD international conference on Management of data (SIGMOD '09)*, pages 19–30. ACM, 2009.

[68] Eric Metois, Paul Yarin, Noah Salzman, and Joshua R. Smith. FiberFingerprint identification. In *Proc. 3rd Workshop on Automatic Identification*, pages 147–154, 2002.

[69] John L. Murphy. Loudspeaker design tradeoffs. `http://www.trueaudio.com/st_trade.htm`, 1997.

[70] Mikhail Myagkov, Peter C. Ordeshook, and Dimitri Shakin. *The Forensics of Election Fraud: Russia and Ukraine*. Cambridge University Press, 2009.

[71] Vishvjit S. Nalwa. Automatic on-line signature verification. In *Proceedings of the Third Asian Conference on Computer Vision-Volume I - Volume I*, ACCV '98, pages 10–15, London, UK, 1997. Springer-Verlag.

[72] Radford M. Neal. Software for low density parity check codes. `http://www.cs.utoronto.ca/~radford/ldpc.software.html`, February 2006.

[73] C. A. Neff. Election confidence: A comparison of methodologies and their relative effectiveness at achieving it, December 2003. `http://www.votehere.net/papers/ElectionConfidence.pdf`.

[74] Elaine Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying facial images. *IEEE Transactions on Knowledge and Data Engineering*, 17:232–243, 2005.

[75] National Institute of Justice. Bullet comparison and identification: Physical characteristics. http://www.nij.gov/training/firearms-training/module11/fir_m11_t04_01.htm.

[76] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[77] Harry F. Olson. *Music, Physics and Engineering*. Dover Publications, 1967.

[78] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & systems (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[79] George Orwell. *1984*. Signet Classic, January 1961.

[80] European Parliament. Directive 2002/58/EC: Data Protection in the Electronic Communications Sector, July 2002. http://europa.eu/scadplus/leg/en/lvb/l24120.htm.

[81] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In *In Aspects of Visual Form*, pages 443–451. World Scientific, 1997.

[82] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-24, Microsoft Research, April 1998.

[83] Victor Reinoso. Memorandum Entitled: Report on DC CAS Testing Security Protocols. http://s3.documentcloud.org/documents/73991/day-three-documents.pdf, September 2009.

[84] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

[85] R. G. Saltman. Effective use of computing technology in vote-tallying. Technical Report NBSIR 75-687, National Bureau of Standards, March 1975.

[86] Jimmy Lee Shreeve. Art forgers: What lies beneath. *The Independent*, Sep. 3, 2008. http://www.independent.co.uk/arts-entertainment/art/features/art-forgers-what-lies-beneath-917067.html.

[87] Jonathan Shultz. Senators criticize onstar for proposed changes to privacy terms. *The New York Times*, 2011. http://wheels.blogs.nytimes.com/2011/09/27/senators-criticize-onstar-for-proposed-changes-to-privacy-terms/.

[88] Sony and Phillips. CD Digital Audio Specification. Technical report, Sony and Phillips, 1999.

[89] C.O.S. Sorzano, P. Thevenaz, and M. Unser. Elastic registration of biological images using vector-spline regularization. *IEEE Trans. Biomedical Engineering*, 52(4):652–663, April 2005.

[90] Ronald D. Spencer. *The Expert versus the Object: Judging Fakes and False Attributions in the Visual Arts*. Oxford University Press, 2004.

[91] Michael Squier. Typewriter evidence; alger hiss' appeal in court may depend on the credibility of a mute witness. *The New York Times*, page SM53, 1952.

[92] H. Stanislevic. Random auditing of e-voting systems: How much is enough?, August 2006. `http://www.votetrustusa.org/pdfs/VTTF/EVEPAuditing.pdf`.

[93] Ken Steiglitz. *A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music*. Prentice Hall, 1996.

[94] Supreme Court of the United States. Certiorari United States v. Jones, January 2012.

[95] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, pages 155–168, New York, NY, USA, 2011. ACM.

[96] *BBC News*. Ticket site closed on fraud fears. `http://news.bbc.co.uk/1/hi/entertainment/7680814.stm`, Oct. 21, 2008.

[97] Floyd E. Toole. Loudspeakers and rooms for multichannel audio reproduction (part 3). Technical report, Harman, January 2002.

[98] U.S. Department of Commerce. Top 10 ways to protect yourself from counterfeiting and piracy. `http://www.stopfakes.gov/pdf/Consumer_Tips.pdf`.

[99] U.S. Department of Health & Human Services. IRB guidebook. `http://www.hhs.gov/ohrp/irb/irb_guidebook.htm`.

[100] Jennifer Valentino-Devries. 'Stingray' Phone Tracker Fuels Constitutional Clash. *Wall Street Journal*, September 2011. `http://online.wsj.com/article/SB10001424053111904194604576583112723197574.html`.

[101] Verified Voting Foundation. The Verifier. `http://www.verifiedvoting.org/verifier/`.

[102] Avery L. Wang. An Industrial-Strength Audio Search Algorithm. In Sayeed Choudhury and Sue Manus, editors, *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, pages 7–13, `http://www.ismir.net`, October 2003. The International Society for Music Information Retrieval, ISMIR.

[103] Wisconsin Government Accountability Board. Spring 2011 election results. `http://gab.wi.gov/elections-voting/results/2011/spring`.

[104] Don Woligroski. Objective benchmarks: Frequency response. http://www.tomshardware.com/reviews/pc-speaker-2.1-channel-subwoofer,2835-8.html, January 2011.

[105] R.J. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intesity. In *Proc. 22nd SPIE Annual Technical Symposium*, volume 155, pages 136–143, 1978.

[106] Z. Xia, S. A. Schneider, J. Heather, and J. Traoré. Analysis, improvement and simplification of Prêt à Voter with Paillier Encryption. In *Proc. 2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '08).*

[107] Davide Zanetti, Boris Danev, and Srdjan Capkun. Physical-layer identification of UHF RFID tags. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 353–364, New York, NY, USA, 2010. ACM.

[108] Kim Zetter. Serious error in diebold voting software caused lost ballots in california county. *Wired.com*, 2008. http://www.wired.com/threatlevel/2008/12/unique-election/.

[109] Baoshi Zhu, Jiankang Wu, and Mohan S. Kankanhalli. Print signatures for document authentication. In *Proc. 10th ACM Conference on Computer and Communications Security*, pages 145–154, 2003.

[110] E. Zwicker. Subdivision of the audible frequency range into critical bands. In *Journal of the Acoustical Society of America*, volume 2, 1961.