ABSTRACT

Title of Dissertation:	SEARCHING HETEROGENEOUS DOCUMENT IMAGE COLLECTIONS
	Rajiv Jain, Doctor of Philosophy, 2015
Dissertation directed by:	Dr. David Doermann, Institute for Advanced Computer Studies

A decrease in data storage costs and widespread use of scanning devices has led to massive quantities of scanned digital documents in corporations, organizations, and governments around the world. Automatically processing these large heterogeneous collections can be difficult due to considerable variation in resolution, quality, font, layout, noise, and content. In order to make this data available to a wide audience, methods for efficient retrieval and analysis from large collections of document images remain an open and important area of research. In this proposal, we present research in three areas that augment the current state of the art in the retrieval and analysis of large heterogeneous document image collections.

First, we explore an efficient approach to document image retrieval, which allows users to perform retrieval against large image collections in a query-by-example manner. Our approach is compared to text retrieval of OCR on a collection of 7 million document images collected from lawsuits against tobacco companies. Next, we present research in document verification and change detection, where one may want to quickly determine if two document images contain any differences (document verification) and if so, to determine precisely what and where changes have occurred (change detection). A motivating example is legal contracts, where scanned images are often e-mailed back and forth and small changes can have severe ramifications. Finally, approaches useful for exploiting the biometric properties of handwriting in order to perform writer identification and retrieval in document images are examined.

SEARCHING HETEROGENEOUS DOCUMENT IMAGE COLLECTIONS

By

Rajiv Jain

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor David Jacobs, Chair
Dr. David Doermann
Professor Doug Oard
Professor Larry Davis
Professor Dana Nau

Acknowledgements

I want to first thank Dr. David Doermann for introducing me to document image processing, advising me over the last 5 years, and being patient with me as I learned to balance my academic schedule while working full time. I appreciate that he always encouraged me to pursue new ideas and made himself available to discuss research any time. I also want to thank Professor Doug Oard for introducing me to the field of retrieval when I first took his class and continuing to guide me in my research afterwards. I enjoyed learning from Professor David Jacobs and it was his Biometrics class that first sparked my interest in writer identification. I would also like to thank Professor Larry Davis and Professor Dana Nau for reviewing my dissertation and agreeing to serve on my committee.

I owe a debt of gratitude to my managers at work, Brad Kline and Christine Edwards, for encouraging me to pursue my education, providing research guidance early in my career, and allowing me to take time off to finish up my research. I want to acknowledge my co-workers Thao Tran, Dallin Akagi, Dave Smith, Adam Wolfe, and Terry Adams for allowing me to bounce research ideas off of them and the many helpful discussions along the way. I would also like to thank my colleagues in the LAMP lab Jayant Kumar, Peng Ye, Le Kang, Sungmin Eum, and Varun Manjunatha for their support.

Finally I would like to recognize my parents, Bhavi and Kala Jain as well as my wife Rachita Varma-Jain for their never-ending support and patience as I juggled work and school with my family.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In the last thirty years, a combination of faster computing, cheaper storage, increased bandwidth, and widespread use of scanning devices has led to massive growth in the number and size of scanned document collections as home users, organizations and governments continue to digitize materials that traditionally resided on paper. Documents such as forms, contracts, bills, memos, and official correspondences are often printed in hard copy format and kept as official records in corporate and government settings. Recent litigation cases against companies [1], [2] and government officials [3] accused of wrongdoing, have also led to the creation of millions of document images for legal departments to manually sort through and organize.

Concurrently, there are massive efforts underway by organizations and governments throughout the world [4], [5], [6], [7] to digitize historical documents, newspapers, magazines and books by scanning and saving them, thus allowing widespread public access to printed materials that would otherwise only be accessible on site. For example, recent estimates by the United States National Archives indicated that it currently holds over 10 billion documents in its collection, containing historical records from the U.S. government spanning the past 400 years [4]. Many of these are of great interest to historians, policy makers, universities, and the general public. Nevertheless, the fact that these documents are in image form makes them difficult to access using traditional information retrieval techniques.

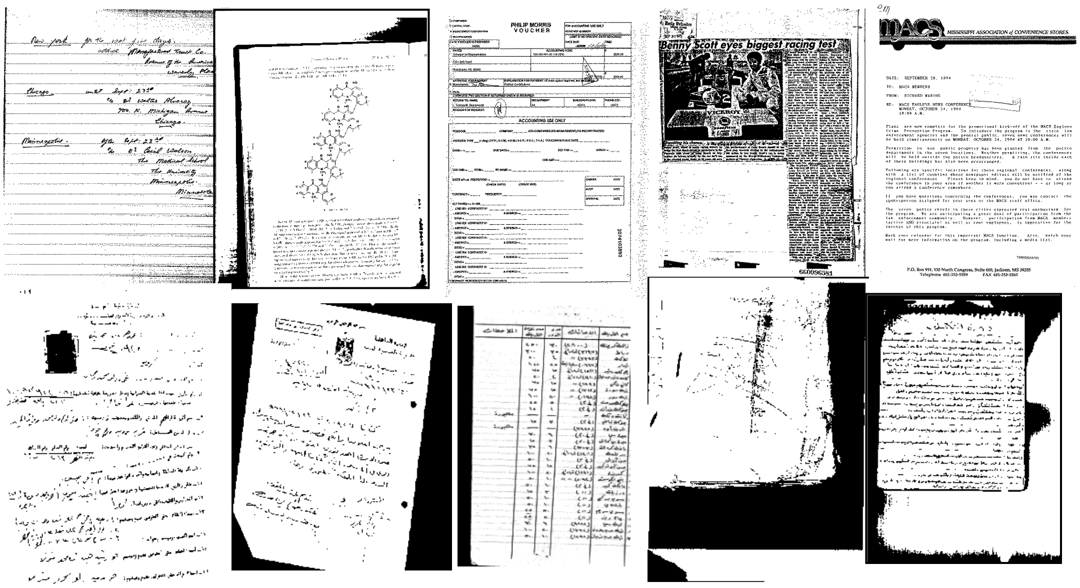
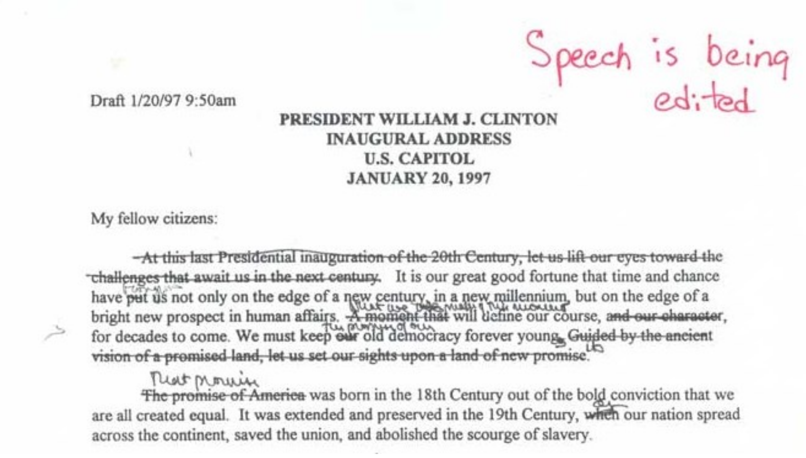

PHILIP MORRIS
VOUCHER
Benny Scott eyes biggest racing test
MACS MISSISSIPPI ASSOCIATION of CONVENIENCE STORES

Figure 1: Examples of varied documents present in large document image collections

Ideally, we would like to build automated systems, with efficient and accurate algorithms, that allow users to quickly find and analyze documents of interest. While conversion to electronic form may work in some situations, these large datasets pose challenges that make this unrealistic. This includes considerable variation in resolution, quality, font, layout, and noise, as well as an eclectic mix of content containing handwriting, forms, photographs, charts, graphs, and signatures as shown in Figure 1.

This dissertation presents work in three areas that augment the current state of the art in the retrieval and analysis of large heterogeneous document image collections. The first topic is **document image retrieval**, which allows users to perform retrieval against large image collections in a query-by-example manner to find visually similar sub-images. The second topic explores **document verification and change detection**, where one may want to quickly determine if two documents contain any differences (document verification) and if so, to determine precisely what and where the changes have occurred (change detection). This is useful in both the filtering and summarizing of retrieval results. Finally, the third research area uses biometric properties of handwriting to perform **writer identification and retrieval** on document images. The remaining sections of this chapter introduce these three topics.



Speech is being edited

Draft 1/20/97 9:50am

PRESIDENT WILLIAM J. CLINTON
INAUGURAL ADDRESS
U.S. CAPITOL
JANUARY 20, 1997

My fellow citizens:

~~At this last Presidential inauguration of the 20th Century, let us lift our eyes toward the challenges that await us in the next century.~~ It is our great good fortune that time and chance have ~~put~~ us not only on the edge of a new century, in a new millennium, but on the edge of a bright new prospect in human affairs. ~~A moment that~~ will define our course, ~~and our character~~, for decades to come. We must keep ~~our~~ old democracy forever young. ~~Guided by the ancient vision of a promised land, let us set our sights upon a land of new promise~~.

That promise ~~The promise of America~~ was born in the 18th Century out of the bold conviction that we are all created equal. It was extended and preserved in the 19th Century, ~~when~~ our nation spread across the continent, saved the union, and abolished the scourge of slavery.

Figure 2: Draft of President Clinton's 1997 Inaugural speech containing his handwritten annotations

To further motivate these three research areas, imagine the scenario of a historian researching the decision process of President Clinton by analyzing records kept from the administration at his presidential library. President Clinton is famous for often editing his own speeches and the historian is hoping that draft edits containing his handwritten annotations as shown in Figure 2 might give an unfiltered insight into his thought process on important policies. The historian can easily access draft versions of speeches without the handwritten edits, but unfortunately the records containing the handwritten edits are not well organized and millions of scanned images would have to be searched through, which would take years to analyze individually. Hence, the historian uses a document image retrieval system to search for near duplicates of each of the speeches. The system then uses an analytic for change detection between the queried and retrieved versions of the draft to automatically isolate the handwritten annotations or changes from the constant machine print text. It turns out that President Clinton was not the only one making handwritten edits to these speeches, but that several of his advisors would also make similar annotations. Thus the historian turns to a writer identification analytic to only retrieve annotations consisting of similar handwriting style to those belonging to President Clinton.

1.1: Document Image Retrieval

Traditionally, the most widely used approach for providing access to document image collections has been to leverage the enormous amount of research in text retrieval by first using Optical Character Recognition (OCR) algorithms to convert the scanned document images to text. There are several problems with such approaches. First, OCR techniques, even if they are perfect, often do not fully capture visual clues such as the font, style (bold, italics, etc.), page layout (genre), table structures, or identify non-text graphical objects such as logos that can help address a user's information needs.

Second, even when documents are primarily text, retrieval algorithms have been optimized for "clean" text. However, OCR algorithms are error prone, with substantial variations in accuracy even with state of the art commercial algorithms, due to factors such as script complexity, noise, resolution, and page layout. While some techniques exist to deal with noisy text, even the best text retrieval techniques begin to break down as the character accuracy rate falls below 75-80% [1].

A large portion of traditional document image research has focused on improving OCR performance, but more recent research has focused on using images in either classification or retrieval applications to perform image retrieval against the pixel representation of the content. Recent progress in image retrieval and increasing computational power has made it possible to scale to datasets with millions of document images. While the information retrieval community has been using collections of this size for decades now, pixel-based document image retrieval research has traditionally focused on much smaller sets. Moreover, image retrieval research has tended to focus on designing algorithms optimized for specific tasks such as logo recognition or page layout analysis, with the implicit assumption that such capabilities would be useful in a more global pipeline.

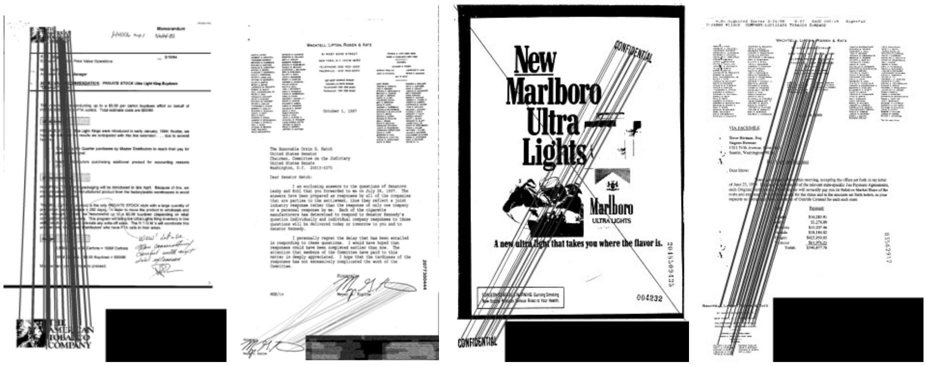

New Marlboro Ultra Lights
Marlboro
ULTRA LIGHTS
A new ultra light that takes you where the flavor is.
CONFIDENTIAL

Figure 3: Examples showing SURF extraction and matching for graphical objects, signatures, text and stamps

Chapter 2 presents a large scale, segmentation-free image retrieval algorithm that indexes local features and uses geometric verification to efficiently search millions of images. The approach is first tested on a logo dataset and experiments demonstrate that the approach can accurately match graphical logos and images of text regions as shown in Figure 3. We then directly compare the image retrieval algorithm to standard text retrieval on OCR'd data on a large real world collection in an attempt to answer the question, "Is image retrieval useful for general document retrieval applications?" Experiments were conducted using topics generated by lawyers, for which relevance judgments are available using seven million document images obtained from lawsuits against tobacco companies.

Our primary contributions include:

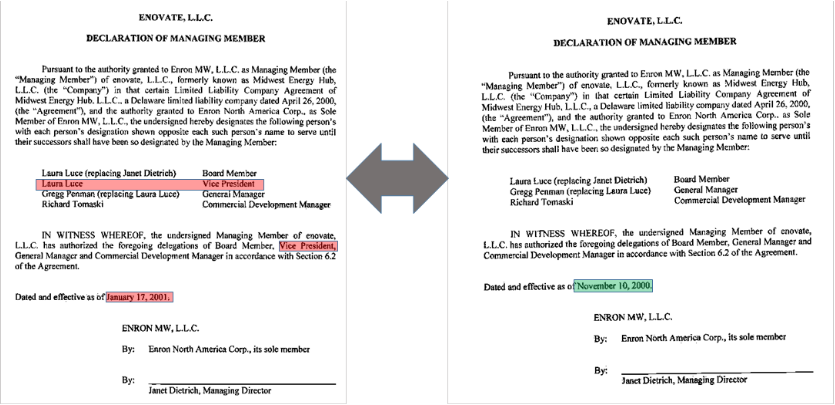
- A scalable, segmentation-free approach to retrieval of text block and graphical objects in document images [8].
- The first study to directly evaluate an image query-by-example technique for user relevance on a large real world dataset [9].

1.2: VisualDiff: Document Verification and Change Detection

The second topic explores the related problems of verification and change detection in document images to determine if two images differ, and if they do, to determine precisely what content may have been added, deleted, or otherwise modified. As with our retrieval work, this is accomplished in the image domain, without the need for complete conversion to electronic form. A motivating example for these capabilities comes from the results of large-scale document image retrieval systems like the one discussed in Section 1.2. These systems often return a list of similar results, and document verification could be used to cluster or suppress identical results. Change detection, on the other hand, could be used as a way to trace the revision history from similar results or could be used to quickly highlight changes of importance from a given query document.

Document images of contractual agreements ("contracts") provide another motivating example for these capabilities. During a contract negotiation process, it is typical for modified versions of the contract to be emailed or faxed back and forth multiple times, often in a scanned image format, prior to a final copy being signed by all parties. Small, undetected changes inadvertently or maliciously introduced prior to signing could lead to severe legal or financial repercussions. Professional contract administrators currently verify changes between two versions of a scanned contract by manually comparing both documents line by line, which is both time consuming and subject to human error.

The goal of document verification is to provide a Boolean decision as to whether the content of two documents is indeed identical, even if the pixel values and image sizes are different. There are many factors which can cause large differences in pixel values between two identical document images. For documents scanned using a traditional flat bed or autofed scanners, common reasons for these changes include 2D rotations, lossy compression, resolution changes, binarization, and embedded enhancement algorithms. Many mobile devices now also include scanning apps that allow users to take pictures of their documents in lieu of a scanner. Even though these popular apps often provide image enhancement algorithms to remove affine or perspective distortion of the image, this form of scanning can introduce more difficult forms of distortion such as 3D pose, lighting, motion blur, and slight warping from curved surfaces since the “scanning” is no longer done in a controlled environment. Noise can also come from the document itself in the form of creases, stains, and ink bleed.



ENOVATE, L.L.C.

DECLARATION OF MANAGING MEMBER

Pursuant to the authority granted to Enron MW, L.L.C. as Managing Member (the "Managing Member") of enovate, L.L.C., formerly known as Midwest Energy Hub, L.L.C. (the "Company") in that certain Limited Liability Company Agreement of Midwest Energy Hub. L.L.C., a Delaware limited liability company dated April 26, 2000, (the "Agreement"), and the authority granted to Enron North America Corp., as Sole Member of Enron MW, L.L.C., the undersigned hereby designates the following person's with each person's designation shown opposite each such person's name to serve until their successors shall have been so designated by the Managing Member:

Laura Luce (replacing Janet Dietrich)	Board Member
Laura Luce	Vice President
Gregg Penman (replacing Laura Luce)	General Manager
Richard Tomaski	Commercial Development Manager

IN WITNESS WHEREOF, the undersigned Managing Member of enovate, L.L.C. has authorized the foregoing delegations of Board Member, Vice President, General Manager and Commercial Development Manager in accordance with Section 6.2 of the Agreement.

Dated and effective as of January 17, 2001.

ENRON MW, L.L.C.

By: Enron North America Corp., its sole member

By: ______________________
Janet Dietrich, Managing Director

ENOVATE, L.L.C.

DECLARATION OF MANAGING MEMBER

Pursuant to the authority granted to Enron MW, L.L.C. as Managing Member (the "Managing Member") of enovate, L.L.C., formerly known as Midwest Energy Hub, L.L.C. (the "Company") in that certain Limited Liability Company Agreement of Midwest Energy Hub, L.L.C., a Delaware limited liability company dated April 26, 2000, (the "Agreement"), and the authority granted to Enron North America Corp.. as Sole Member of Enron MW, L.L.C., the undersigned hereby designates the following person's with each person's designation shown opposite each such person's name to serve until their successors shall have been so designated by the Managing Member:

Laura Luce (replacing Janet Dietrich)	Board Member
Gregg Penman (replacing Laura Luce)	General Manager
Richard Tomaski	Commercial Development Manager

IN WITNESS WHEREOF, the undersigned Managing Member of enovate, L.L.C. has authorized the foregoing delegations of Board Member, General Manager and Commercial Development Manager in accordance with Section 6.2 of the Agreement.

Dated and effective as of November 10, 2000.

ENRON MW, L.L.C.

By: Enron North America Corp., its sole member

By: ______________________
Janet Dietrich, Managing Director

Figure 4: Change detection example for 2 similar documents. Deletions in red; Additions in green.

In cases for which document verification indicates a difference, the goal of change detection is to assist a user in determining the precise differences between two documents as shown in Figure 4. There are many modifications that can cause large changes to the appearance of a document. These range from very basic formatting changes (font size or style, margins, line spacing, etc.) to complex rewriting of the content. For most applications the addition or deletion of content, including text, graphics, or handwriting such as signatures and notes is of primary importance. Users may also want to detect when the content is identical but the layout of the page, line spacing, font style, size and/or color of text has changed. The work presented in Chapter 3 constrains the world of possible changes to additions and deletions of content.

Our primary contributions include:

- A document verification algorithm to determine if two document images contain identical content across common image transformations [10].
- A change detection approach to detect word level changes by applying the longest common subsequence algorithm to local features extracted from text line images [11].

- A segmentation-free change detection technique to detect word or character level changes even if page or line segmentation fails [11].

1.3: Writer Identification and Retrieval

Handwriting is a behavioral biometric, which has been used for over one thousand years to identify the authors of unattributed pieces of handwriting. An example of variation present in samples from the same writer as well as their distinctiveness in relation to different writers is shown in Figure 5. Even with the decreasing popularity of handwriting for daily use, writer identification and retrieval remains a relevant and important research area for law enforcement agencies. For example, in 2010 the FBI used handwriting analysis to identify and arrest an individual who mailed threatening handwritten letters containing white powder to political leaders. Worldwide there are estimated to be several hundred forensic handwriting experts, the most widely known group is the Questioned Documents Unit of the FBI, which compares handwriting samples from questioned documents (i.e. ransom notes, death threat letters, or fraudulent official documents) to samples taken from suspects in order to identify perpetrators. Forensic examiners estimated it takes 2 investigators approximately 2-3 days to adequately complete a case involving approximately a dozen documents. Cases that involve several thousands of handwritten documents would overwhelm Forensic examiners, and the hope is that an automated or even semi-automated solution would help assist these experts.

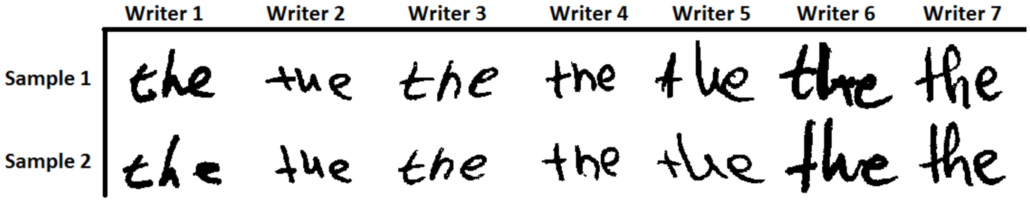


Figure 5: Examples of two handwriting samples written by several different individuals

There are applications for research in writer identification and retrieval beyond law enforcement as well. Paleographers, who study ancient handwriting, are often interested in identifying all the manuscripts from a certain scribe because a scribe often uses the same vernacular constructs and abbreviations throughout their writings, which make it easier for a Paleographer to translate at one time. One could also imagine school officials wanting to ensure that a student's handwriting is consistent throughout exams and homework assignments to ensure another individual had not taken their place.

One final motivating example comes from optical character recognition of large volumes of handwriting, which remains an unsolved and challenging problem for the document image community due to large variations present in the shapes and styles between writers. Recent research has shown that handwriting models adapted to a particular writer do significantly better than general models [12], [13]. One could imagine writer identification being used to match a new handwriting sample to a pre-trained OCR engine that contained the closest writing style.

Chapter 4 presents two approaches using local features for writer identification. The first method extracts adjacent line segments from the contour of connected components and improves upon existing edge based features. The second approach introduces a more general framework for writer identification that attempts to mimic an approach taken by forensic examiners. Repeatable character-like segments are extracted to form a pseudo alphabet and these pseudo-characters are described using an improved contour based gradient descriptor. We also explore using the more powerful Fisher Vector [14] for feature pooling and combinations local features to produce state-of-the-art results. These methods are validated on datasets and contests consisting of hundreds of writers across several languages and shown to produce state of the art results.

Our primary contributions include:

- Applying the K-Adjacent Segments (KAS) feature to writer identification. [15]. This is the strongest edge-based feature used for writer identification.
- An automated, general framework for writer identification inspired by allograph matching performed by handwriting forensic examiners. [16]
    - This method won the ICDAR 2013 Writer ID Contest [17].
- Weighted combinations of local features including KAS, Contour Gradient Descriptors, and SURF to produce state-of-the-art results [18].

1.4: Dissertation Overview

The remainder of the thesis is organized by topic, with each topic in a separate chapter. Document image retrieval, change detection, and writer identification are presented in Chapters 2, 3, and 4 respectively. Each of these chapters includes related work, approaches, experimental evaluations, and conclusions with open areas for future research. Chapter 5 summarizes our contributions and publications.

Chapter 2: Document Image Retrieval

In this chapter, we explore the problem of document image retrieval in an image query-by-example context with the intent of creating a general technique that can be applied to large heterogeneous collections. A scalable, segmentation-free document image retrieval approach is introduced that can accurately retrieve sub-images of graphical objects and structured text blocks using only one query image. We first demonstrate the effectiveness of our approach in an experiment designed for logo retrieval. Next, we scale our algorithm to 7 million document images collected from tobacco lawsuits with the goal of studying user relevance in a more general information retrieval setting.

2.1: Related Work

The most common method of information retrieval on document images continues to be text retrieval on the output generated from OCR programs. This is a popular option due to the efficiency of text retrieval systems as well as the increasing speed and precision of OCR technology. However, OCR still suffers from varying degrees of inaccuracy depending on the language, so strategies have been researched in the past to cope with different levels of OCR degradation. In the past decade, techniques were developed that allow image retrieval researchers to begin performing query by example image retrieval for graphical objects within a document image. Active research supporting this includes word spotting, page layout analysis and logo retrieval especially when OCR error rates are too high to perform adequate text retrieval. Surprisingly, little work has been done directly comparing the text retrieval against the benefits of image retrieval techniques.

2.1.1: Information Retrieval Basics

The field of information retrieval (IR) is a user-centered discipline in which computer algorithms are designed to efficiently examine vast volumes of content in order to satisfy a user's information needs. While the field originated in the library sciences in order to support librarians in quickly retrieving reading material for patrons, the Internet has helped expand the field from books to other forms of content such as web pages, music, video, and images. In the traditional IR model a user formulates their information needs into a query that can be input to a system and the system generates a set of ranked results for review by the user. The notion of relevance, which describes results that satisfy the user's information need, is central to the field. From a computer science perspective, the goal is to create a general purpose retrieval algorithm that can return all relevant items at the top of the ranked list in real time. In order to measure the effectiveness of an information retrieval system, several evaluation measures have been devised, the most basic of which are precision and recall. Precision, shown in equation (1), measures the percentage of returned results that are relevant. Recall, shown in equation (2), measures the percentage of relevant documents returned relative to the number of relevant documents in a dataset.

$$\text{Recall} = \text{R(k)} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \tag{1}$$

$$\text{Precision} = \text{P(k)} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \tag{2}$$

When using a ranked list, both precision and recall need to be calculated at various ranks to generate a plot in order to gain a complete picture of how a system is performing. In response, two single value metrics were created, the F-measure and Average Precision. The F-measure, shown in equation (3), balances between precision and recall and can be thought of as the harmonic mean.

$$\mathrm{F-measure} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{3}$$

Average precision is one of the most common information retrieval metrics and can be calculated by equation (4) where n is the number of results, P(k) is the precision at rank k, rel(k) is 1 if result k is relevant and 0 otherwise, and R is the number of relevant documents in the dataset. Mean Average Precision (MAP) is the mean of the average precision across all queries, Q.

$$\mathrm{Avg\ Precision} = \mathrm{AvgP} = \frac{\sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{P(k)} * \mathrm{rel(k)}}{\mathrm{R}} \tag{4}$$

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AvgP(q)}}{|\mathrm{Q}|} \tag{5}$$

2.1.2: OCR Text Retrieval

In 2007, the Tesseract [19] OCR system, which had been acquired by Google, open-sourced their OCR platform and began publishing a series of papers [20], [21], [22] providing insight into the inner working of general commercial OCR systems. The basic flow for converting a document image to text is shown in Figure 6. The intent of this diagram is not to focus on the inner workings of Tesseract, but rather to show the many areas an OCR application can be susceptible to mistakes. Errors in the page and line segmentation or skew correction could cause large blocks of text to be entirely missed. This commonly happens for document images with non-standard layouts. Errors in the word segmentation, character segmentation, or character classification often cause errors at the word level.

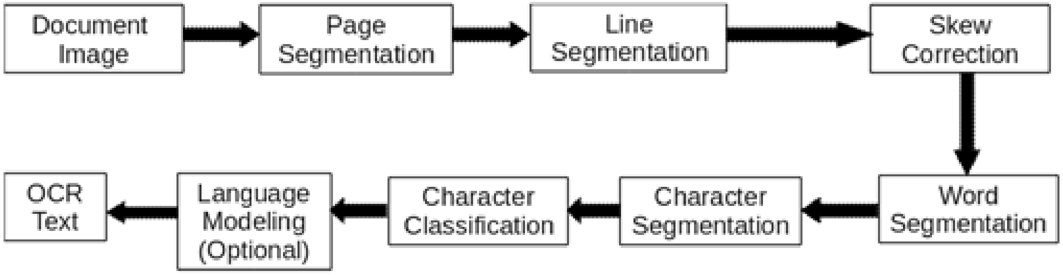


Figure 6: Algorithm flow for a general OCR system such as Tesseract.

Table 1 shows reported character and word error rates for various languages from various OCR systems. Please note that these results often come from experiments with ideal test data. Studies on large real world data-sets have shown OCR retrieval systems may have word error rates of greater than 50% on 1/3 of all English document images [23] even though academic and commercial systems commonly report less than 1% character error rates with English.

Language	Character Error Rate (CER)	Word Error Rate (WER)
English	0.5% [21]	3.72% [21]
Hindi	15.41% [21] , 8.7% [24]	69.44% [21]
Chinese	3.77% [21]	N/A
Arabic	N/A	14.1% [25]
Arabic, Handwritten	N/A	30% [26]

Table 1: Table showing various error rates for current state of the art OCR

The work of Doermann [27] and Beitzel et al. [28] provides an overview of recent OCR error correction and retrieval techniques that are tolerant to the character and word errors. Please note that no text based retrieval algorithm can handle errors where entire regions of text are missing, as is common with poor page segmentation. The most common first step has been to use natural language processing techniques to fit the OCR output into a probabilistic language model such as a finite state transducer to choose the most likely sentence [29]. While obtaining training data is relatively easy, this technique is limited by its dictionary and can often have trouble with out of vocabulary terms such as uncommon names or other pronouns. Another approach taken by Kolak et al. [30] creates a noisy channel model specific to the OCR algorithm and calculates the probability that the OCR algorithm will make certain character errors, given a ground truth training data mapping common OCR errors in the model. Experiments have shown the WER decreases from 20% to 5% using this approach on OCR text from the French Bible.

Given blocks of OCR text with varying levels of accuracy, past research has also examined how to best index text for efficient and fault tolerant retrieval. As a general rule, prior research on OCR shows that 1) for character accuracy between 70-80% character n-gram techniques perform well, 2) for 80-95% accuracy, enhanced IR techniques work well, and 3) most vector-space retrieval algorithms are only tolerant on OCR above 95% accuracy [27]. In [31] and [32] Taghva et al. show the vector space retrieval model is largely unaffected by simple OCR errors, especially when there is a large amount of content present in the data-based documents. Using simple OCR correction techniques, they were able to recover most documents that were retrieved due to OCR errors, except in cases with very low OCR quality or large segmentation errors. The work by Harding et al. [33] used character n-grams to perform retrieval on OCR'd text and showed that it significantly outperformed traditional retrieval techniques with character error rates greater than 10%. Recent work by Bulco-Neto et al. in [34] showed that Latent Semantic Indexing (LSI) can be used to retrieve documents with OCR errors since the surrounding context for misspelled words is often similar. Hassann et al. [35] on the other hand used Latent Dirichlet Allocation (LDA) to perform retrieval based on the topic model for a given document and demonstrated that OCR with a 22% CER can be performed using this technique.

2.1.3: Document Image Retrieval

Document image researchers have begun actively exploring methods for retrieval by using an image as a query in order to address cases where the OCR is unavailable or visual features are more descriptive than the textual content. There have been three main areas of research in document image retrieval, which are covered in the sections below. The first section discusses page layout analysis where documents with similar structures are matched. The second section reviews past approaches for retrieval of graphical objects such as logos or signatures. Finally the third section discusses keyword spotting of distinct symbols and words, which is useful when other approaches for OCR fail.

2.1.3.1: Page Layout

The original purpose of page layout analysis was to break a page into zones that could be fed into an OCR engine. Early work by O'Gorman focused on simple layouts such as those found in magazines, books, and journals [36]. More recently Kise et al. proposed an approach for non-Manhattan layouts which found zones by creating a Voronoi diagram around connected components on the page [37] that was further extended by Agrawal and Doermann for complex and handwritten material [38]. Page layout analysis was also one of the earliest techniques for performing document image retrieval without using the OCR text because the structure of a document image can sometimes provide just as much information as the textual content. For example, when looking for official memos or forms from a company that always has the same structure, one can key off of the layout of the page even though the textual content can change. A full survey of this page layout research can be found in [27] and [39]. Much of the early work on the topic focused on using a document's layout to perform genre classification [27], [40]. In recent work, Huang et al. proposed a retrieval algorithm in [41] that compared quadrilaterals formed from lines on the pages from two documents to determine whether two documents have similar layouts. They achieved a MAP of 0.7 on a 2855 document dataset. In [42], Gordo and Valveny used a cyclic polar description of text zones in a page to create a rotation invariant descriptor for a page. Experiments demonstrated a MAP of 0.6 on a dataset of 823 Spanish government documents. Marinai et al. [43] directly compare the trees from XY-cut page segmentation to determine page similarity. Their experiments on 22,253 pages extracted from 53 books demonstrate that documents with similar structures are clustered together. In [44] Nakai et al. introduces Locally Likely Arrangement Hashing (LLAH), to return near duplicate images that are invariant to affine translations from camera pictures. He uses the center of word features as interest points and describes each point with geometric relationships to neighboring points rather than the actual content. More recently in [45] Takeda et al. scale the algorithm to ten million images, though the index is required to reside in memory and requires over 150 GB. Results show that he is able to find the same document across pose changes with 92% precision viewing as little as 1/8th of the document.

2.1.3.2: Graphical Objects

In recent years, there have been a number of papers exploring the related topics of detection, recognition, and retrieval in document images of graphical objects such as logos and signatures, which cannot be handled by OCR. Given a document image, detection can be defined as the problem of finding a graphical object's boundary on the page without regard to class. Recognition (or matching) on the other hand is the problem of determining to which class a given logo or signature belongs. Retrieval can be viewed as a combination of the two problems where one wants to efficiently and simultaneously detect and recognize a graphical object across a large dataset given some query image.

Doermann et al. presented one of the first approaches for logo retrieval on document images in [46]. He first performs logo detection on zones from a page using texture features based on wavelets and then performs logo recognition using shape descriptors based on algebraic and differential invariants. Logo detection was more recently explored by [47], [48] and [49]. In [47], Zhu and Doermann detect logos on a page using connected component features and a Fisher classifier. Wang and Chen use a decision tree to grow rectangle boundaries around candidate logos in [48]. In [49], Li et al. use local descriptors found using difference of Gaussians and described using connected component features to detect logos. In [50], Rusinol and Llados explore efficient logo retrieval on logos by indexing shape context descriptors and achieves 82.6 mean average precision (MAP) on the Tobacco 800 dataset. Zhu et al. extend their detection work in [51] to build a retrieval system and performs recognition by matching local shape context descriptors, reporting a MAP score of 82.6%. The closest work to ours has been done by Rusinol and Llados [52], who perform logo retrieval using a bag of SIFT features. He reports a true positive rate of 90.2 % and a false positive rate of 1%, but the experiments are done on a different dataset that is not publicly available making direct comparison difficult. Similar approaches have been used for retrieval of graphical structures on a page such as engineering drawings [53].

While there is a long history of work on signature verification and identification [54], more recent research has focused on performing document image retrieval based on handwriting signatures. In [55] Srihari et al. assume that a signature has already been extracted and designs a retrieval system that removes machine text noise from the signature and uses gradient, structural and concavity features to perform retrieval across a large dataset on signatures present in document images. He achieves a precision of 89.6% and recall of 88.6% when comparing the top 10 results on a dataset of 447 signatures. Agam et al. extend this work in [56] by combining text retrieval with signature retrieval on the CDIP Tobacco dataset to show that signatures could be tied to certain attributes such as the amount of money a tobacco CEO handled. While there was little experimentation, as far as we know this is one of the only other studies to examine the relationship between text retrieval and document image retrieval of graphical objects using a large real world dataset. Zhu et al. creates a signature detection algorithm [57], which takes advantage of the distinct attributes of a signature by employing a multi-scale approach that looks for salient regions based on the curvature of a given connected component. Signature retrieval is then performed using a shape matching algorithm that compares the relative positions of points sampled between two signatures.

2.1.3.3: Keyword Spotting

In cases where general purpose OCR algorithms fail and the font and script has little variation within a dataset, document image researchers have employed techniques that match the image of a word directly against a document image. Rath and Manmatha had one of the first successes using this approach in noisy handwritten historical manuscripts [58]. A word segmentation algorithm was used to extract all the words from the page and then an exemplar word is matched against all the words on a page using contour and gradient features. The average precision of this algorithm is 72%, but a downside is that the algorithm is limited by the accuracy of the page and word segmentation. More recently Rusinol et al. proposed a segmentation free word spotting algorithm [52], built on their earlier work in [59]. He uses dense SIFT features in a bag of features framework, which is similar to the approach used in Chapter 3. The features are mapped to code words and then represented using Latent Semantic Indexing. During the retrieval, a document image is scanned over several scales for areas with many similar patches and candidate patches are verified geometrically. He reports a MAP of 42% and demonstrates that the algorithm works across a large variety of fonts and languages.

2.2: Segmentation Free Document Image Retrieval

2.2.1: Local Feature Extraction

Our use of local descriptors was motivated by the work of Ke et al. [60], which showed excellent results for the near duplicate image retrieval problem when using the SIFT descriptor. One can imagine retrieval in document images as an extension of the near duplicate image retrieval problem in computer vision, where one wishes to find all similar images that could have been created from simple image transformations such as cropping, scaling, or rotating. Thus local features that are scale and rotation invariant are desirable for logo retrieval because of their ability to match sub sections of images with these transformations. Large affine translations are not a concern for document image retrieval since most large collections contain images that are created by scanning on a flat surface. Local feature extraction for images can be split into two steps: interest point detection and feature description. A good interest point detector extracts patches from an image that are distinct and repeatable across common image transformations such as scaling, rotating, and cropping. An illustration of these patches can be found in Figure 7. Next, each patch is represented by a feature vector, which ideally captures the shape and texture of the pixels within the patch, but is invariant to noise and variations that occur across similar images. Document images are especially challenging because the pixels are often binary, meaning that there is little texture information and a substantial amount of noise present in images from the binarization process.

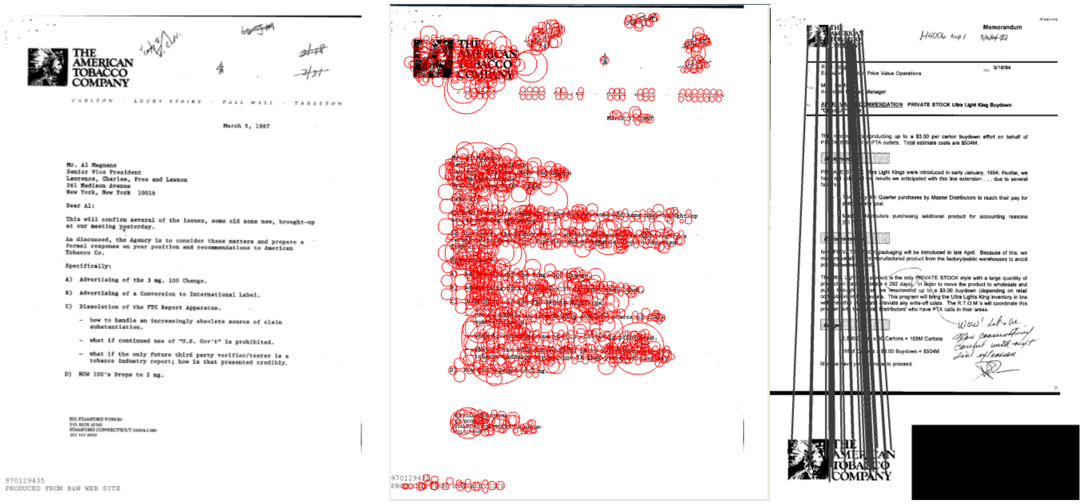


Figure 7: Example showing SURF extraction and matching

Speeded Up Robust Features (SURF) [61] were chosen for this work because it has shown good performance for image retrieval, is fast to compute, and is more resilient to noise than other popular local features such as SIFT [62]. SURF uses the fast Hessian interest point detector, which finds patches with the largest high gradient change in comparison to neighboring patches and in scaled space. The SURF feature descriptor for a given patch is calculated by first equally subdividing a given patch into a 4x4 grid. For each subsection, the Haar wavelet response Dx and Dy are computed in the x and y directions respectively. The original SURF descriptor calculates the following four attributes (∑Dx, ∑Dy, ∑ |Dx|, ∑|Dy|) per interest point. However, the first two features ∑Dx and ∑Dy contain little information in binary images. Hence they are excluded to form a more compact 32 dimensional feature vector, which is ¼ the size of the SIFT descriptor, without any loss in accuracy. An open source C++ implementation of the SURF algorithm from the OpenCV software package has been modified to produce the smaller feature vector described above. A more detailed analysis of SURF can be found in the original paper [61].

To build a naïve retrieval system using these descriptors one would first extract SURF features from each document image offline. Then at query time one would extract SURF features from a region of interest in the image and do a pair-wise comparison between all the features extracted from the document and the logo and then choose the document with the most matches. Given that on average 7000 features are extracted from each document, 1000 features are extracted from small regions of interest, and 32 calculations are required for each feature comparison, it becomes quickly apparent that this approach will not scale to datasets with millions of images due to its computational requirements.

2.2.1: Feature Indexing

An indexing technique that maps feature vectors to hash codes in order to build an inverted index was explored to improve query speed. The method used for this study was motivated by a recent indexing technique for near duplicate images [63], which attempts to group feature vectors that are distinct along the same dimensions. The original technique defined the *distinctiveness* D for a given feature vector *v* as:

$$D(i) = |\mathrm{v}_i - \mu_i| * \sigma_i \tag{6}$$

Where μ_i and σ_i are the mean and standard deviation for the distribution of position *i* over the feature vector. The method proposed in that paper did not perform well because the equation they used to quantify the distinctiveness was rewarding, instead of penalizing, dimensions with high variance. Also, the direction of the distinctiveness is lost by taking the absolute value. Instead the following alternative distinctiveness measure using the Z-score from statistics is proposed:

$$D(i) = \frac{v_i - \mu_i}{\sigma_i} \tag{7}$$

Both μ_i and σ_i are computed offline for each of the 32 dimensions in the SURF feature vector using feature vectors from randomly selected documents in the CDIP collection. Two index keys for each feature vector are formed by taking the six positions with the highest distinctiveness as well as the six positions with the lowest distinctiveness score and sorting the two index values numerically. Note there are fewer hashes than the six required for the algorithm presented in [63]. The index is further expanded by using one bit to represent the sign of the Laplacian in the fast Hessian detector and another bit to represent whether the hash came from the highest or lowest distinctness scores. The use of six positions to create the hash is determined empirically since it provides a hash space of 3,624,768 values with a tradeoff of slightly more neighboring feature vectors not being hashed to the same point.

To clarify the indexing procedure, an example of a ten dimensional feature vector and an index made of three positions is given in Table 2. Here the index keys become the positions with the three highest distinctiveness scores (highlighted in blue) and the positions with the three lowest distinctiveness scores (highlighted in yellow) sorted numerically. The first (or high) key is (6, 7, 10) and the second (or low) key is (4, 5, 8).

As with all approximate nearest neighbor algorithms there is no guarantee that two points indexed to the same key truly match. To solve this problem a low dimensional representation of the feature vector is stored along with the index key and verify that an indexed feature vector falls within a given distance threshold of the query at runtime. To minimize the storage cost and computational requirements of this matching, the SURF feature vector is reduced to eight dimensions using PCA. This indexing scheme is used to create an inverted index as follows:

Key 1 -> Doc ID -> X, Y coordinates, Orientation, Feature Vector

Key 2 -> Doc ID -> X, Y coordinates, Orientation, Feature Vector

Each index key points to the unique ID for the document it was computed from and its associated feature vector. The X and Y coordinates, as well as the orientation of the interest point, are stored for geometric filtering. This index reduces search complexity by >10^8 over the naïve approach.

X	1	2	3	4	5	6	7	8	9	10
v_i	5	7	3	2	1	9	8	0	6	10
μ_i	5	5	5	5	5	5	5	5	5	5
σ_i	1	1	1	1	1	1	1	1	1	1
$D(i)$	0	2	-2	-3	-4	4	3	-5	1	5

Table 2: Distinctiveness scores for an example feature vector

2.2.3: Properties of the Index

Figure 8 shows the document frequency of a given hash for a set of 1000 scanned documents and approximately seven million interest points. The hashes clearly follow a power law distribution using local descriptors and this phenomenon has been noticed in previous papers using local features [64]. In this case, the most frequent hashes appear to be associated with straight lines, which occur frequently throughout the dataset. The 1000 indexes with the high frequency are put on a stop wordlist because these points are not discriminative and occur several times in most documents. This removes approximately 20% of the interest points from the index and significantly speeds up query performance since indexes with the largest number of entries take the longest time to load from a disk.

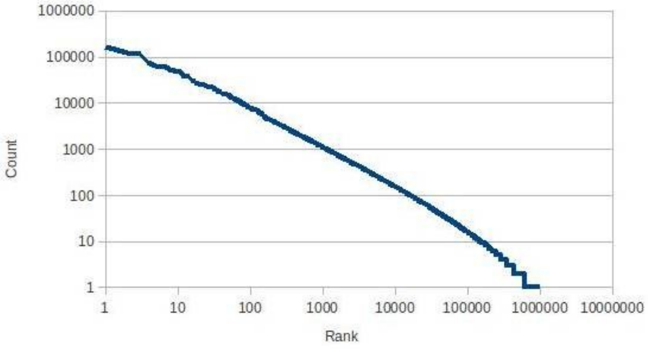


Figure 8: Graph of the index key frequencies sorted by their rank.

This indexing scheme is designed to reside on disk. Each entry in the index is 19 bytes (six bytes for the document ID, four bytes for the X, Y coordinates, one byte for the key point orientation, and eight bytes for the Feature Vector). Thus an average image with 7000 features, each with two entries, requires approximately 266KB of disk space. Once the high frequency hashes are removed, this is reduced to 212KB of disk space per image. This could likely be reduced by half with better bit management and a better dimensionality reduction technique than PCA.

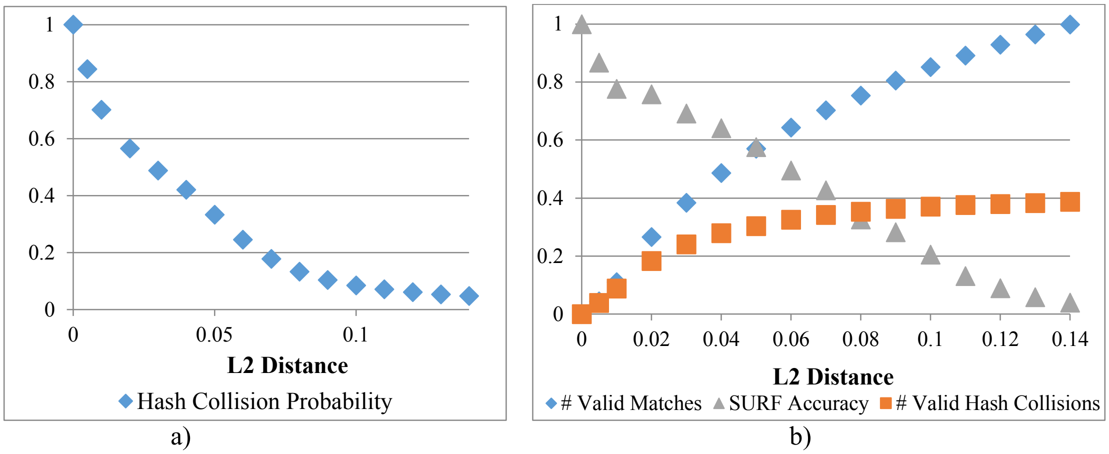


Figure 9: Hash properties given L2 distance between 2 SURF features. a) Probability of a hash collision. b) Given a brute force query with matches marked true or false if they correspond with the correct region: Gray – Accuracy of SURF features. Blue –Accumulating percentage of true SURF matches. Orange – Percentage of true SURF matches that also have hash collisions.

Figure 9 provides an analysis of the hashing technique given the L2 distance between SURF features compared using the brute force method between pairs of images with matching regions. SURF matches were marked true if they linked the correct region between the two images, and otherwise were marked as false. Figure 9a shows the probability of a hash collision given the L2 distance between SURF features. To put these probabilities in context, Figure 9b shows that about 45% of all valid matches are retained until SURF has a false match rate of 80% at a L2 distance of 0.1, and 40% of all valid matches are retained until SURF has a false match rate of 96% at a L2 distance of 0.14. In practice, matches beyond a distance of 0.1 create too large a false positive rate to be useful for retrieval. While a substantial number of SURF matches are lost, experiments show only a small reduction in recall due to the large number of features being extracted per image allowing many opportunities for point matches between corresponding regions. Figure 10 shows an example of the matches found using the hash index in comparison to the Brute force method. Other current state of the art feature indexing approaches such as KD-trees [65], which could potentially provide much higher recall, would also require a very large amount

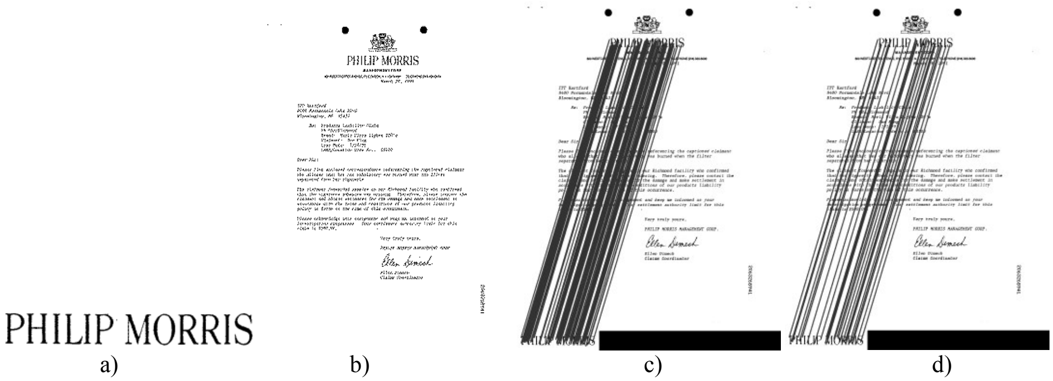


Figure 10: Example Query. a) Query Image. b) Indexed Image. c) Brute Force matches with geometric verification. d) Indexed matches with hash collisions and geometric verification. Note about 50% of matches are lost from 2c.

of RAM to be practical with the volume of features being extracted. The simplicity of this indexing scheme, its large hash space, and ability to allow efficient indexing on traditional hard disks sets this hashing technique apart from other approaches.

2.2.4: Filtering Using Geometric Consistency

Image retrieval systems built on indexing local descriptors have traditionally used RANSAC [66] to perform geometric verification. Others have used Hough transforms [62] for the same purpose because RANSAC performance degrades if a significant portion of matching features are outliers. Since affine transformations are not a priority for scanned document images, a much simpler two-step geometric filter is used. The first step takes advantage of the orientation information provided by interest points found using the fast Hessian detector. The orientation difference of valid matching points between a logo query and document image should be relatively constant and equal to the skew between the images. Thus, the orientation of each query interest point is subtracted from all matching interest points in the database and normalized to fall within 0 and 360 degrees. For a given image with matching interest points, a sliding scale of six degrees is used. Interest points that fall within the window with the largest number of matches are kept and the rest are discarded. In cases of images with erroneously matched interest points this can significantly reduce the error rate. The sliding window is trivial in cost and can be programmed on the order of O(n), where n is the number of matching points. Note how the number of false matches is significantly reduced in Figure 5b.

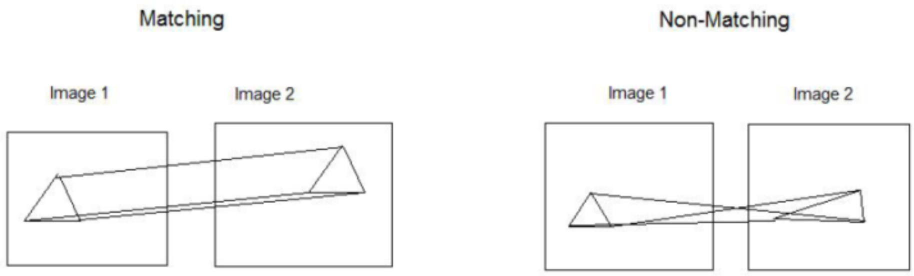


Figure 11: An illustration of the triangle filter.

The second step uses a stricter filter, but with the tradeoff that its computational cost is O(n^3). Triangles are computed from all combinations of three matching points between the query and document image. Given paired triangles in the query and document image, the difference between the corresponding angles is computed. If the angles differ by three degrees, the triangle is ignored. Features that are a part of at least two valid triangles are retained and the final score returned by this step for ranking results is the number of matching triangles. Figure 11 illustrates this triangle filter and Figure 12 shows how these two filters remove false positives.

To limit the effect of a large number of matches on the computation of the triangle filter, the 100 matches with the smallest distances are stored per image before applying the second filter. To reduce the cost of the triangle filter in a large scale implementation, one could randomly sample the set of all triangles. However, in practice this filtering is nominal in cost because there were few erroneous matching points after the first filter was applied, so all triangles were sampled in the implementation. While efficiency is always a concern, the filtering can afford to be more expensive than the feature matching because only the top results need to be verified and this process takes much less time than the index retrieval.

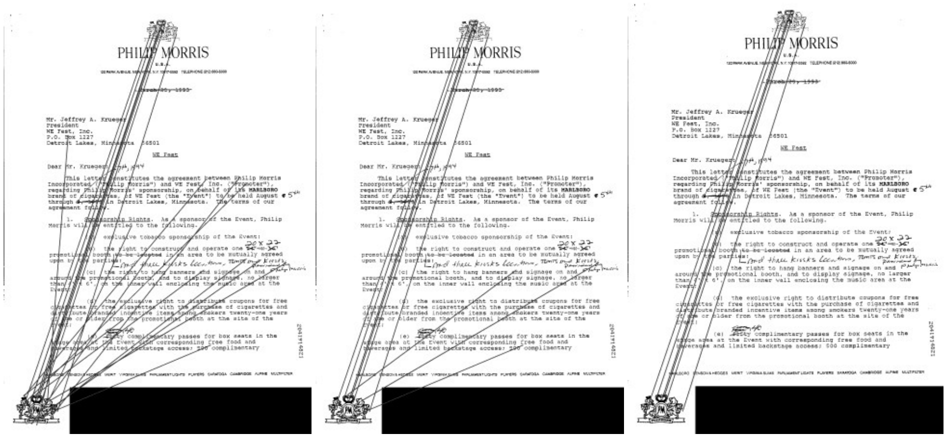


Figure 12: (a) no filters, (b) orientation filter, (c) triangle filter

2.3: Experiments: Logo Retrieval

2.3.1: Dataset

The UMD Tobacco 800 dataset ([1], [56], [67]) is an 800 document/1290 page subset of the CDIP 7 million document/42 million page dataset received after state litigation related to tobacco. All images have been scanned in binary format and range in resolution from 150 DPI to 300 DPI. Figure 13 shows how noisy the images can be as a result of the binarization. It has become the standard public dataset for work on logos in document images. Ground truth labels of the logos were created in ([47], [51]) and only consist of the graphical portion of the logo. The dataset contains 35 unique logo classes across 435 pages. Only logo classes with two or more occurrences are used as query images in experiments. Each image is resized to have a greatest dimension of 2000 pixels or 180 DPI to reduce the number of features generated for images with much higher resolution.

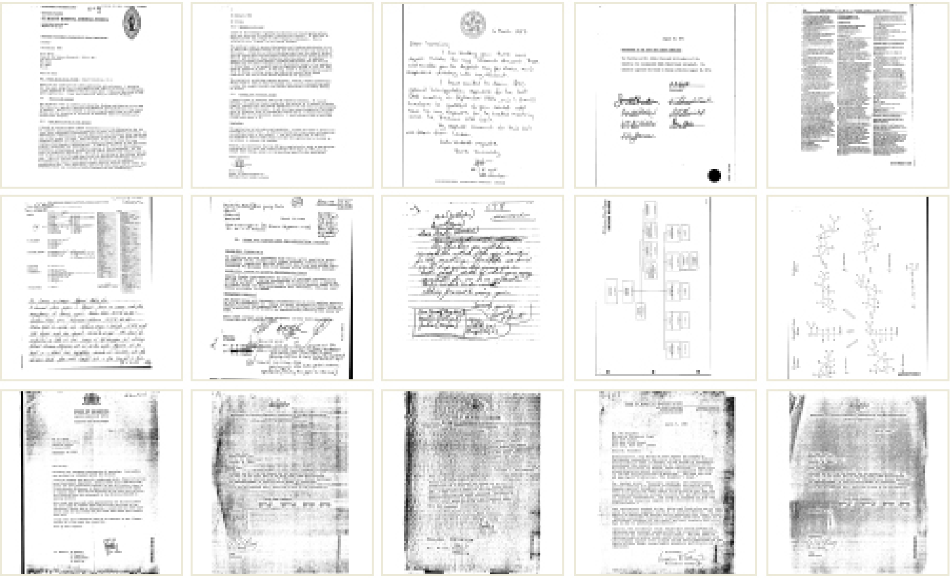


Figure 13: 15 sample pages from the CDIP dataset

2.3.2: Evaluation measures

The score reported in the results for a given system is the mean average precision (MAP), shown in Equation (5), which is the mean of the average precision scores across all queries. A few logos are disproportionately represented in this dataset so the MAP score is also computed by taking the average across all classes as well as all queries. Queries are submitted for each of the ground truth logos provided by [47] against all 1290 pages of the Tobacco 800 dataset. Examples of these logos can be seen in Figure 13.

2.3.3: Results

2.3.3.1: Results on the Tobacco 800 dataset

The following three configurations of the system were tested and the results are in Table 3: Brute force searching, indexed search with geometric verification, indexed search with/without geometric verification. The results using the graphical logo alone were lower than expected. A close inspection of the results showed that the system was operating with high precision for all of the logos, but noisy logos that were heavily impacted by the binarization or small logos that comprised of a small portion of the entire page for which few features were extracted had low recall. One positive result from this data was that there was only a 10-14% drop in the MAP score between the brute force query and the indexed query. Most of this loss was due to a loss in recall from fewer matching points. Another positive result from this data was that the geometric verification significantly improved the results by increasing the MAP by 17-22%. This was largely due to the increase in precision.

System	MAP per logo	MAP per query
Brute force	.67	.59
Index with geometric verification	.57	.45
Index without geometric verification	.35	.28

Table 3: Results on the tobacco 800 dataset for graphical logos

2.3.3.2: What is a logo?

The logo queries chosen from the ground truth of [47] omit contextual text from the logo when possible to limit the test set to graphical objects. However, in reality for each logo there is almost always uniquely identifying titles or text blocks adjacent to the logo that could be used as part of a query image to boost performance. In many cases the text is more consistent, prominent and distinct than the logo. Three more image queries are run on the Tobacco 800 dataset using the indexed search with geometric verification to compare how the contextual text surrounding the logo affects performance: Logo alone, Text[1] alone, Logo + Text Image[1]. The logos are reused from the prior experiment and the Text and Logo + Text images are manually extracted for each page containing logo. The MAP per query and MAP per logo class

[1] In this context, “Text” refers to Images of Text, as opposed to electronic text

are again used as metrics for performance. Examples of the Logo, Text, and Logo + Text images can be found in Table 4.

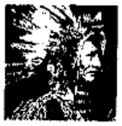
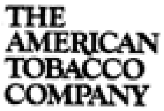
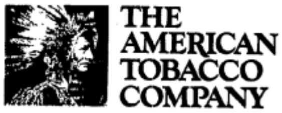
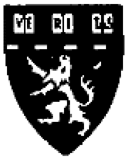
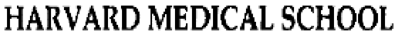
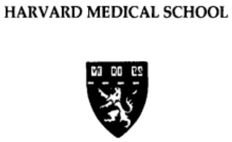
Logo	Text	Logo + text
	 THE AMERICAN TOBACCO COMPANY	 THE AMERICAN TOBACCO COMPANY
	 HARVARD MEDICAL SCHOOL	 HARVARD MEDICAL SCHOOL

Table 4: Examples of the text context found with logos

The results in Table 5 show a significant improvement gained by combining the textual and logo information and indicate that graphical objects should not be isolated from their surrounding context when performing logo retrieval on document images. For some documents, logos contain the most distinctive features and for others, the text surrounding the logos is more distinctive. By combining the two, the image query algorithm benefits from having more information and more descriptors. Since the algorithm operates with high precision, the additional text descriptors do not result in many more false positives. One exception was the “Philip Morris” text image, which found several other document images that contained the words but not the logos.

System	MAP Score per logo class	MAP Score per query
Logo only	.57	.45
Text only	.56	.63
Logo + text	.87	.88

Table 5: Results for graphical and text logos

<u>2.4: Experiments: Large Scale Retrieval for User Relevance</u>

2.4.1 Dataset

The collection used for the experiments is the Complex Document Information Processing (CDIP) test collection [1], which is a superset of the Tobacco800 dataset used in the previous experiments. CDIP includes 7 million scanned documents and over 42 million pages, received from tobacco company lawsuits. All images have been scanned in binary format from many different scanners and range in resolution from 150 DPI to 300 DPI. There are many types of documents in the collection, including research papers, e-mails, letters, memos, books, and handwritten notes. The documents have many nonstandard layouts and often include graphics such as logos, tables, graphs, photos, and signatures. Figure 13 shows how noisy the images can be as a result of the binarization. The CDIP collection also includes English OCR text and annotated metadata for each document. This collection was used for the TREC Legal Track from 2006-2008 [[68], [23]], but the complexity of the scanned documents resulted in poor OCR text quality, making this collection an interesting IR challenge for noisy text. TREC Legal worked with lawyers to create “mock” complaints, over 100 topics, and associated Boolean queries. It was impractical for the TREC team to ground truth the entire dataset, so instead they created relevance judgments by pooling the top results from participating systems and truth marking samples from those results. Topics with fewer than 5 judgments of “relevant” were discarded, leaving 55 topics.

2.4.2 Experimental Design

The goal of the experiments was to compare the technique described in section 2.2 to text retrieval of OCR and determine if it provided any utility in satisfying a user's information need for a set of topics on the CDIP tobacco dataset. The setup of this experiment closely mimics the evaluation of TREC Legal. The hope was that the two modes of retrieval would be complementary and that the image retrieval results would improve retrieval performance on at least a few topics when the results of the two techniques were combined. A more modest goal of the experiment was to see if there was a positive relationship between document image retrieval and increased query performance to show that these techniques could be used in cases where OCR failed and text retrieval was not possible.

Due to limited resources (we used one server to build the index), it was not possible to process, index, and store the 1.5 TB, 42 million page collection for image retrieval. Instead, only the first page for each of the seven million documents from the CDIP collection was used resulting in 40 billion indexed features. Lucene [69] was chosen as the text retrieval system for this experiment since it is very popular both commercially and academically for text based information retrieval. Unlike text retrieval, one major drawback with most image retrieval algorithms is that it is almost impossible for a user to make a query without having an existing image of interest. To overcome this limitation in the experiments, text queries using the words from each of the TREC Legal Boolean queries were run and the text and images corresponding to the top 1000 ranked results were retained. The actual Boolean queries were not used because they do not provide ranked results.

2.4.2.1 Relevance Feedback

The first two experiments evaluated the two retrieval strategies using relevance feedback, where the top N relevant documents that were returned by the initial query were used to resubmit new queries. Rocchio's algorithm, which is commonly used to perform relevance feedback, is shown in Equation (8).

$$\overrightarrow{Q_m} = (\alpha * \overrightarrow{Q_O}) + (\beta * \frac{1}{|D_r|} * \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}) - (\gamma * \frac{1}{|D_{nr}|} * \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}) \qquad (8)$$

Here Q_O is the original query vector, $\overrightarrow{D_j}$ is a related document vector, D_r is the set of relevant documents, and D_n is the set of unrelated documents. $\alpha, \beta,$ and γ are constants used to balance the importance of the relevance feedback results against query drift from the original query. In the experiments, unrelated documents were not used in the relevance feedback so γ was always set to zero. α and β were varied experimentally to determine the optimal performance for relevance feedback and to verify that the relevance feedback does indeed improve performance. When comparing two systems note that $(\alpha * \overrightarrow{Q_O})$ cancels out and that the only metric needed to directly compare both relevance feedback algorithms is the relative ranking using only the feedback from applying:

$$\overrightarrow{Q_{rel}} = \frac{1}{|D_r|} * \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j} \qquad (9)$$

While application of this formula is straightforward for text retrieval by substituting TF-IDF scores, the image retrieval query vectors occupy a different vector space than the original query. Thus instead of applying the formula using the TF-IDF score from each query, the rank of the result was substituted instead. For both experiments three combinations were evaluated: the original query alone, the original query + text relevance feedback, and the original query + image relevance feedback. The image queries were conducted using the algorithm presented in Section 3 with the entire document image result. Document images submitted to the text retrieval system used the text OCR and the entire textual content was used for the text query. The first experiment simulated user relevance feedback by using the top five relevant documents returned from the initial query for relevance feedback. The second experiment simulated blind relevance feedback, where the Top 10 ranked results (relevant or not) were submitted for relevance feedback.

2.4.2.2 User Queries

The third experiment attempted to determine if users could improve image retrieval by only selecting relevant sub-images rather than the entire document image. In order to avoid biases, three different users selected five distinct topics from the collection on which they thought image processing would perform best. They were supplied with all relevant images from the initial query for each TREC topic and asked to select three distinct document images from the collection that they felt were most relevant to the topic. To be fair to both retrieval systems, the users were asked to select the best image sub-region and the best text sub-region. Both of these regions, as well as the full document image, were then submitted to both retrieval algorithms and each was treated as a new ad-hoc query when evaluating the system.

2.4.2.3 Evaluation Measures

Mean Average Precision (MAP) is a widely used evaluation metric that balances precision and recall into a single value metric averaged across all queries.

However, recent studies have shown that MAP, breaks down when used with incomplete relevance judgments. Because the relevance judgments are sparse, we have chosen to report Bpref [70], an evaluation measure optimized for experiments with sparse relevance judgments. Bpref is a relative measurement that works by ignoring documents without relevance judgments and instead measures the number of relevant documents found above non-relevant documents in a ranked list. It is calculated by Equation (10):

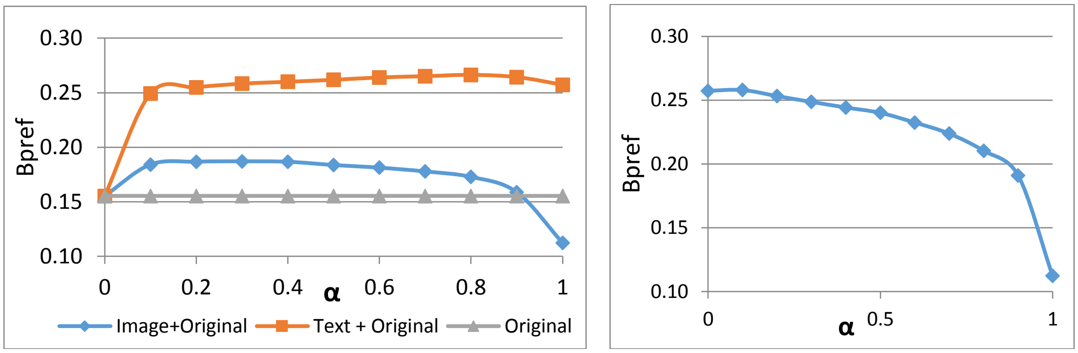
$$\text{Bpref} = \frac{1}{R}\sum_{r}(1 - \frac{|\, n \text{ ranked higher then } r \,|}{\min{(\text{R}, \text{N}\,)}}) \tag{10}$$

Where R is the number of judged relevant documents, N is the total number of judged non-relevant documents, *r* is a retrieved document that is relevant, and *n* is the number of non-relevant documents ranked higher than *r*. Query results of up to a depth of 10,000 were considered due to the sparse number of relevance judgments. The experimental procedure and metrics used are consistent with the experiments done by the TREC Legal Track on this collection [23].

A measurement for precision was also used to complement the Bpref results, especially because we expected image retrieval results to be less well represented in the existing TREC Legal relevance judgments. For that reason, we asked three new assessors to provide relevance judgments for the top 10 results from 20 randomly selected queries, using each system for the three experiments. We used a majority vote to produce a new ground truth from which a traditional precision at 10 measure could be computed. These results, denoted as P(10), are informative as an indication of early precision, but are not directly comparable to the results we report using the sparse TREC Legal relevance judgments.

2.4.3: Results

2.4.3.1: Simulated Relevance Feedback



a) *b)*

Figure 14: Simulated Relevance Feedback Results. a) Retrieval improvement using text/image relevance feedback. b) Retrieval performance of the combined image+text relevance feedback with varying weight (α)

Both image and text retrieval approaches positively impact query performance when the top relevant results were resubmitted for relevance feedback. The graph in Figure 14a shows the Bpref score in order to examine the relationship between the original query and the relevance feedback results. The image relevance feedback is optimal with a weighting of α=.7 and β=.3, which provided an average improvement of 20% over the original query and improved 40 of the 55 queries. The text relevance feedback is optimal with a weighting of α=.2 and β=.8, which provided an average improvement of 71% over the original query and improved 46 of the 55 queries.

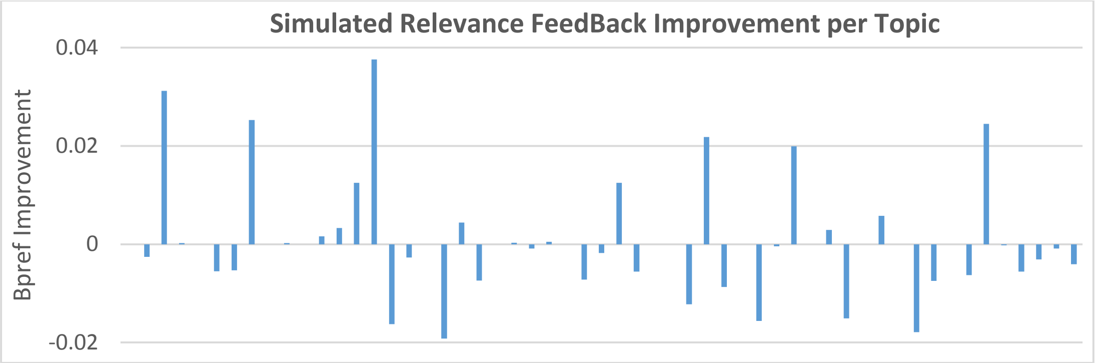


Figure 15: Relevance feedback improvement of image+text retrieval over text retrieval: topics ranked worst to best based on text retrieval performance

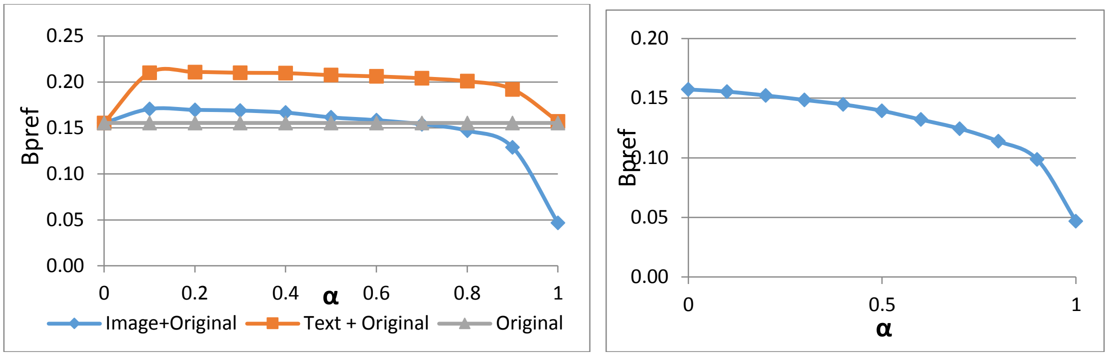
Since both image and text retrieval positively impact relevance feedback performance, the question is now whether image + text retrieval is better than text retrieval alone. The results in Table 6 show the Bpref and Precision scores averaged across all queries. When compared independently, the text retrieval outperforms the image retrieval. Figure 14b displays the Bpref score for various weights of image and text retrieval. The image + text retrieval combination only improves the results by 0.25% mainly because there is a substantial amount of overlap between the positive matches in the two results sets. Hidden from the graph is the fact that image retrieval outperformed text retrieval on 4 queries and image+text retrieval outperformed text retrieval on 17 queries.

Feedback	Bpref	P(10)
Text	0.25	0.66
Image	0.11	0.44

Table 6: Bpref and Precision at 10 results for the simulated relevance feedback.

The CDIP topics were not built with image retrieval capabilities in mind, and thus even modest improvement of the image retrieval system on a few queries is sufficient to indicate the potential of these algorithms in retrieval settings. One advantage of an image retrieval system is that it can work on degraded documents and in cases where the OCR fails. To test this hypothesis, the results are sorted in ascending order based on the text retrieval Bpref score and the Bpref improvement from the image + text retrieval is measured in Figure 15. The results show that several of the first 15 topics with lowest text retrieval performance have substantial improvement by combining the text and image retrieval results.

2.4.3.2: Blind Relevance Feedback



a)

b)

Figure 16: Blind Relevance Feedback Results. a) Retrieval improvement using text/image relevance feedback. b) Retrieval performance of the combined image+text relevance feedback with varying weight (α)

The results for BRF follow the same pattern as the previous section. The graph in Figure 16a shows that both image and text retrieval approaches positively impact query performance when the top 10 results were resubmitted for blind relevance feedback with the text retrieval again outperforming the image retrieval. The blind relevance feedback for image queries is optimal with a weighting of α=.9 and β=.1 and provides an average improvement of 10% over the original query. The text relevance feedback is optimal with a weighting of α=.2 and β=.8 and provides an average improvement of 36% over the original query. The image retrieval improved 29 of 55 queries and the text retrieval improved performance 40 of the 55 queries.

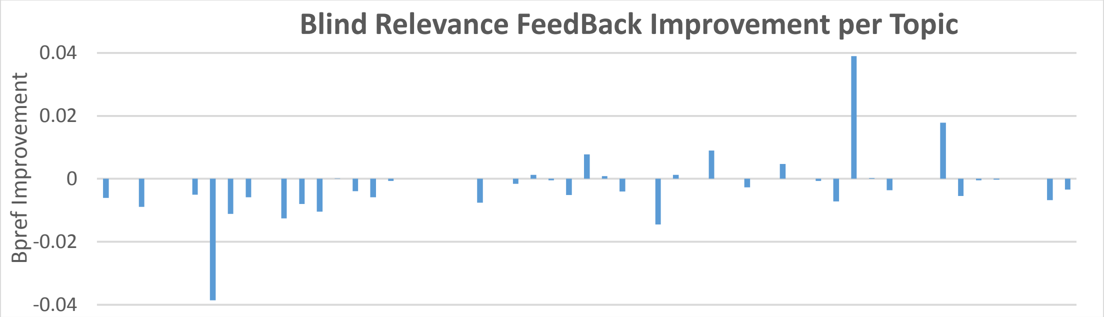


Figure 17: Relevance feedback improvement of image+text retrieval over text retrieval: topics ranked worst to best based on text retrieval performance

The results in Table 7 also show that the precision was lower than the simulated relevance feedback and that the precision of the image retrieval was again lower than the text retrieval. This could be due to the poor performance of the original query. In many cases only one or two of the top ten documents used for BRF were actually relevant to the topic. Image retrieval in general is error prone and the fact that it was not more adversely impacted by the presence of non-relevant documents is surprising.

Feedback	Bpref	P(10)
Text	0.15	.57
Image	0.04	.34

Table 7: Bpref and Precision at 10 results for the blind relevance feedback.

Figure 16b shows that text retrieval outperforms any combination of text retrieval and image retrieval on average across the datasets. However, the image retrieval outperforms the text retrieval on two queries and the combined image+text retrieval performs better than text retrieval on 8 of 55 queries. The improvement of the combined image+text retrieval over text retrieval is again examined in Figure 17 for cases where the text queries performed poorly. The results are mostly negative with results appearing to somewhat improve in a couple of the bottom cases, but they also appear to get substantially worse for most of the other queries.

2.4.3.3: User Queries

The results in Table 8 show the Bpref scores for the text, image, and combined retrieval for the text region, image region selected by the user as well as for the entire page. Even though the image region is labeled with the word "Image" the regions selected by users contained at least some text in almost all cases. For many of the documents selected by users, there were no graphical objects and thus users struggled to select a region and often chose unique parts of the page layout. Unlike the relevance feedback results, each of the three image queries were treated as new ad-hoc queries, which is why the results may seem lower in comparison to the other two studies. The combination of both retrieval techniques was tried for various weights similar to the relevance feedback experiments and the optimal weighting scheme is shown in Table 8. Image+Text Bpref performance improved by a modest 3.3% for image region queries.

Bpref results for user queries			
	Text	**Image**	**Entire Page**
Text	0.159	0.092	0.181
Image	0.038	0.037	0.046
Text+Image	0.159	0.095	0.180
% Change	0%	3.3%	-0.5%

Precision at 10 results for user queries			
	Text	**Image**	**Entire Page**
Text	0.69	0.45	0.80
Image	0.45	0.45	0.49
Text+Image	0.71	0.49	0.80
% Change	2.8%	8.5%	0%

Table 8: Bpref and precision at 10 retrieval results for user selected regions

This is also reflected in the P(10) results in Table 8, which shows an 8.5% improvement in precision for the image region and 2.8% improvement for the text region. This suggests that there is relevant content in the image region that is not available to the OCR. The difference in relative scores between the Bpref scores and P(10) results (a factor of 4 compared to a factor of 2), also suggests that the relevance judgments are biased against image retrieval as relevant documents were likely not included in the judgment pools, lowering the Bpref scores for image querying. The image retrieval on its own outperformed text retrieval on four queries when evaluating the text region. It also outperformed text retrieval on three queries when evaluating the entire page, and 14 queries when evaluating the image regions. When the image retrieval was combined with the text retrieval it outperformed the text retrieval on seven queries for text region, eight queries for the entire page, and 15 queries for image regions. This improvement on a limited number of queries also suggests that the image retrieval may be beneficial in some unique cases when prominent visual features exist in an image. The use of sub-regions did not help query performance for either technique, likely because less information was available to the algorithms. This was the first time users had tried the image query paradigm and one explanation for the drop in performance is that they were unable to select the best documents or regions for optimal image retrieval.

2.4.3.4: Impact of Poor OCR

The accuracy or quality of an OCR system is typically expressed using character and/or word error rates. However, in the absence of a substantial amount of ground truth test data, both of these values are difficult to accurately measure. The work in [71] and [23] used a more ad-hoc measure known as OCR Score, which gives a rough estimate of the word error rate. OCR Scores can be calculated by counting the number of 4+ letter words in the page that appear in a dictionary and dividing it by the total number of 4+ letter words in a page. One of the major advantages of image retrieval over text retrieval is that it is not dependent on OCR output and thus hypothetically better handles poor quality document images. To evaluate this hypothesis, the top 1000 ranked results from the text and image retrieval were evaluated to determine the frequency for various ranges of OCR Scores.

OCR Score	100-95	95-90	90-80	80-70	70-60	60-50	50-25	25-0
Image Retrieval	14.3	27.5	28.7	11.3	3.42	1.98	3.08	9.56
Text Retrieval	22.2	37.8	31.5	6.6	1.4	0.4	0.1	0

Table 9: Comparison of image and text retrieval OCR Scores

The results in Table 9 show image retrieval returns substantially more documents at an OCR Score of 80% or lower. Results with an OCR Score of 25% or lower make up about 10% of the image retrieval results even though not a single result with an OCR Score this low was returned by the text retrieval system. Unfortunately, there were few relevance judgments on document images with poor OCR quality because the pooling of results in the tobacco collection was based on the top results from text retrieval systems that entered the TREC competition. Even, the TREC study in [23] showed approximately 33% of the CDIP collection had an OCR score below 50% and text retrieval approaches in the study had difficulty retrieving documents from this subset of the collection. This made it difficult to assess how beneficial the image retrieval would be in these cases when OCR is likely to fail. The OCR score was calculated for the judgments provided by our assessors for the P(10) calculation in order to determine whether this same phenomenon was seen in the limited results available. In this case, the average OCR score for the text results was 0.89, the average OCR score for the image results was 0.81, and the distribution was similar to Table 9, providing further evidence that the text retrieval favors documents with high OCR quality.

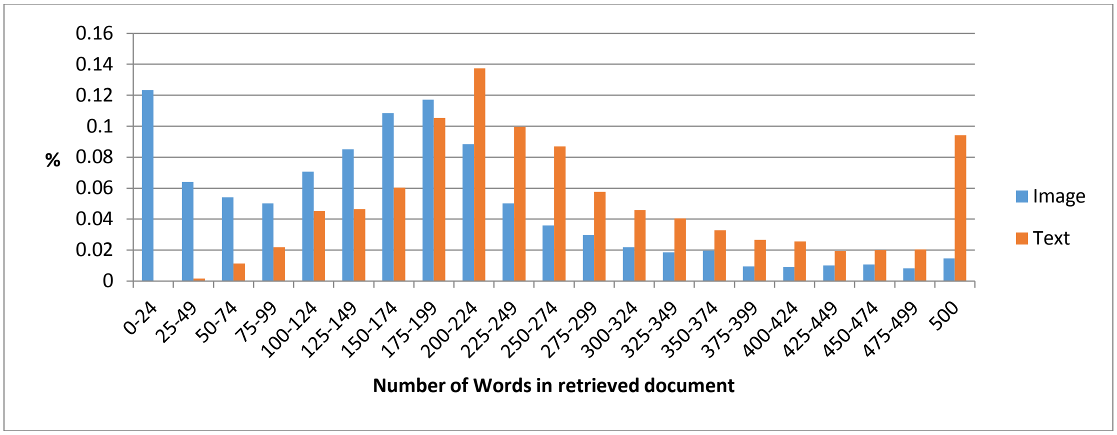


Figure 18: Percentage of documents retrieved with a given word count for both retrieval systems.

While OCR Score is a great measure for studying the effect of the word error rate on retrieval performance, it will not accurately reflect the effect of segmentation errors, where large portions of the page do not have OCR. Instead, this would be manifested by the image retrieval results having fewer words per result. The top 1000 ranked results are again taken from both retrieval systems and this time the percentage of results returned are shown for various word counts in Figure 18. The average word count for the image retrieval is far lower at 167, while the average OCR for the text retrieval is 287. The fact that 19% of the image retrieval results has less than 50 words, while only .1% of text retrieval results did, indicates that that there may be relevant information on the page that the OCR is unable to extract. While image retrieval may not always be needed when OCR quality is good, these results indicate that recall may be increased if image retrieval technology is used for documents with a low OCR score or when few words are extracted from a page. Since systems that participated in the Legal Track used text retrieval of OCR or Metadata, the pooled relevance judgments are possibly biased towards the capabilities of these systems, meaning that documents with little text or poor OCR quality were less likely to have been evaluated.

2.4.3.4: Further Analysis

The results from simulated relevance feedback were analyzed further in order to provide greater insight into the performance of the image retrieval system. In order to give tangible examples the types of queries and results that were generated, Figure 19 and Figure 20 include a text description of the topic (from CDIP), the initial text query (from CDIP dataset), query images used for relevance feedback, and ranked results from image and text retrieval for two of these topics. Topic 78 was chosen because the relevance feedback results using image retrieval outperformed text retrieval by 2% using the Bpref metric. Similarly, Topic 13 was chosen since the text retrieval results were far superior to the image retrieval results with a 35% increase in Bpref. As the results in Figure 19 and Figure 20 show, the image retrieval results were generally near duplicates of the query images at the highest ranks. At lower ranks, the image retrieval results show that the system primarily matched prominent sub-images of the query images such as the US Patent header, the Lorillard logo, or the law offices header and address block.

In the case of Topic 78, which was looking for any documents related to patents of odors, the visual representation of the US Patent Header was important because it is found on all patent submissions and the image query therefore brought back a significant number of relevant patents. This is in contrast to the text retrieval results, which brought back a large number of studies referencing odors, but did not bring back a significant amount of documents also referencing patents. While the OCR did pull out “United States Patent” correctly for text retrieval, the visual importance of the header given its size, font, and location is not conveyed in the OCR text used in the relevance feedback.

Topic 13, which focuses on documents related to chocolate or candy cigarettes, presents a case in which the image retrieval system can fail if the visually prominent portions of the images are not relevant to the topic. This can also occur in cases where there is little content for the image retrieval system to match, such as a document containing pure text with no repeatable visual patterns in common with other relevant documents. Even though, the first couple of image retrieval results for this topic were relevant near duplicates, the majority of results afterwards largely contained either the Lorillard logo or the Brumbaugh law header. Since documents from these companies occurred frequently in the CDIP collection and candy cigarettes were a very small portion of their work, the vast majority of the documents returned were about the business of these companies rather than the topic of interest. The text retrieval results on the other hand focused on chocolate or candy cigarettes since the terms appear often in the query documents and likely had low document frequency raising their prominence in the TF-IDF bag of words model. Unfortunately, none of the CDIP topics centered on individual companies or people, where matching sub-images such as the headers, logos, or address blocks like in the examples above would have likely done very well. Most of the topics were focused on general illegal actions taken by all tobacco companies such as hiding harmful side effects, selling cigarettes to kids, or bribing officials, which were difficult for the image retrieval system to find repeating visual patterns relevant to the topic.

Topic 78

Description: All documents referencing patents on odors, excluding tobacco or cigarette related patents

Initial Query: patent* odor* NOT (tobacco OR cigarette)

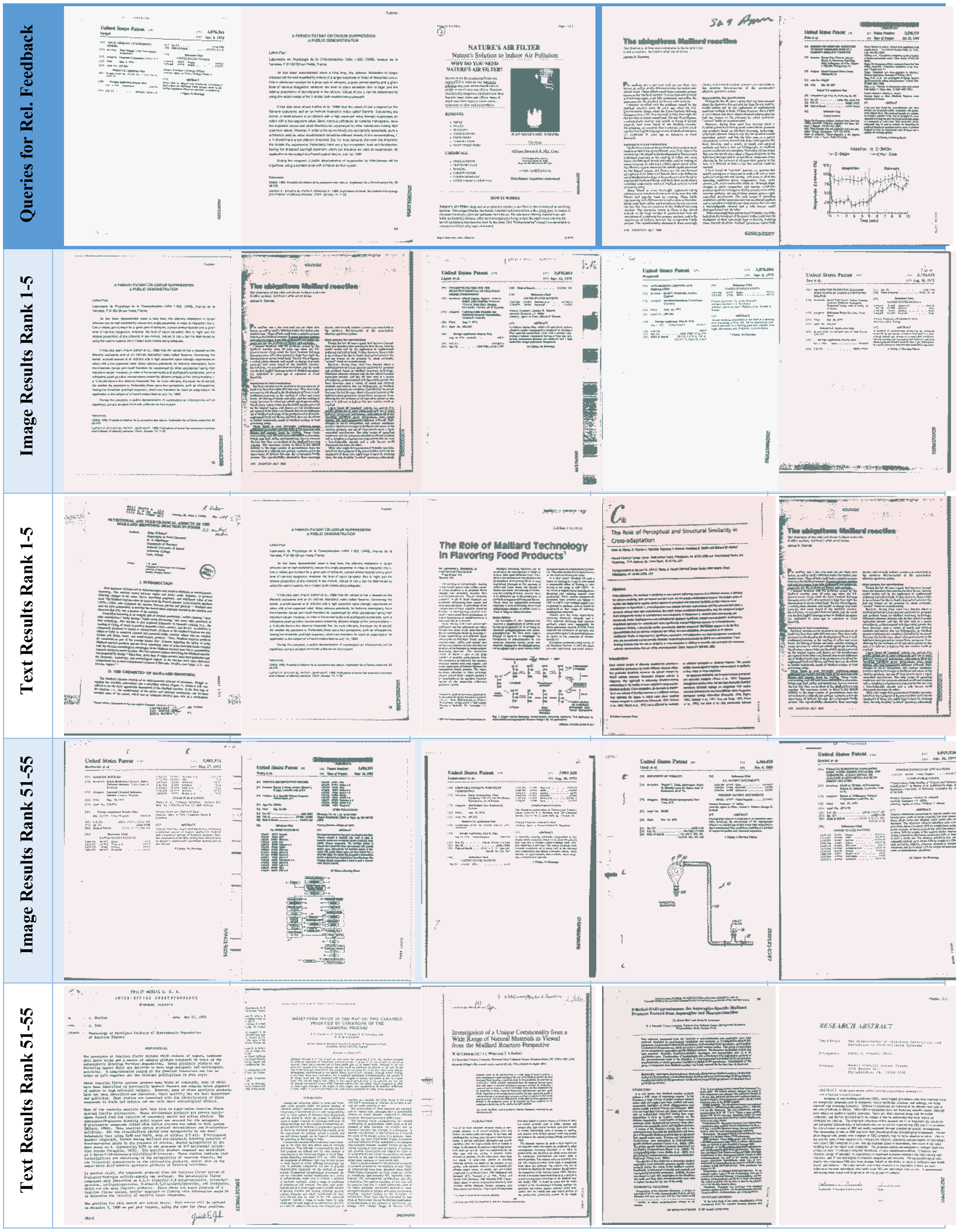


Figure 19: Image and Text Retrieval Results for Topic 78. The first row contains the first five relevant documents returned from the initial query, which are used to perform relevance feedback. The next 4 rows contain ranked results from image and text retrieval.

Topic 13

Description: All documents to or from employees of a tobacco company or tobacco organization referring to the marketing, placement, or sale of chocolate candies in the form of cigarettes.
Initial Query: cand* chocolate cigarette*

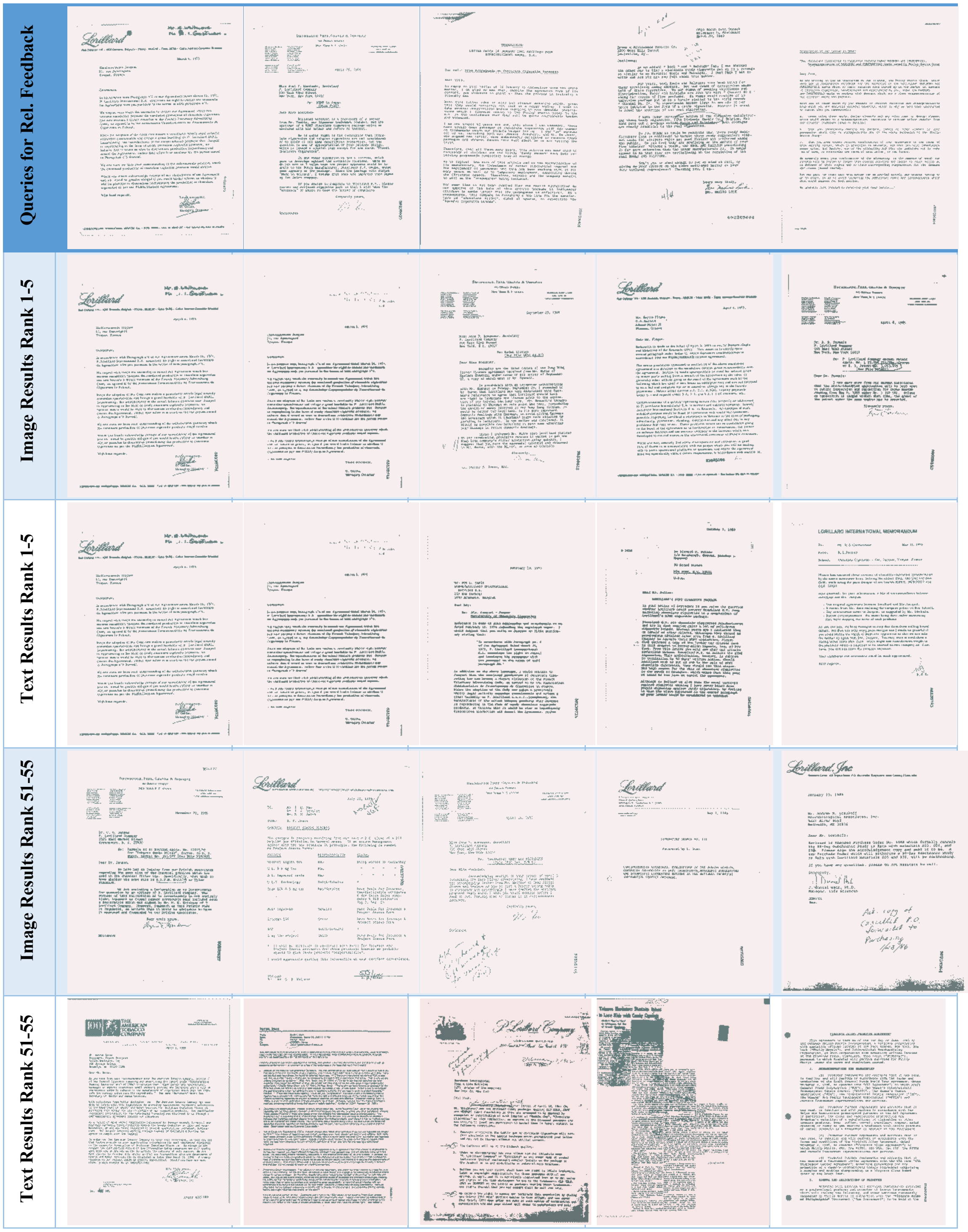


Figure 20: Image and Text Retrieval Results for Topic 13. The first row contains the first five relevant documents returned from the initial query, which are used to perform relevance feedback. The next 4 rows contain ranked results from image and text retrieval.

2.4.3.5: Efficiency

In order for an image retrieval algorithm to be useful it must scale to large numbers of images on commodity hardware and allow for modest indexing and retrieval times. Using a grid computing engine with 400 nodes, it took approximately eight hours to extract on average 7000 SURF features per page and index all seven million document images using the techniques described in Section 2.2. The resulting index was two terabytes in size, though more efficient use of disk space could easily reduce the index to one terabyte. While this is substantial in size, unlike many other image retrieval techniques, this algorithm is able to achieve reasonable search times with the index residing entirely on a hard disk, which is trivial in cost when compared to RAM. The algorithm was designed to have the index distributed across a large number of hard disks using a distributed database such as HBase or residing on a solid state drive to reduce the impact of random seek and disk read time when making thousands of index lookups. Due to limited resources, the index was loaded on a single server and spread across 8 disks. For typical image region queries like text blocks, titles, or logos, the average query time across all seven million images was about 13 seconds.

Algorithm	Feature Size	Index Size	Feature Extraction	Disk Access	Feature Comparison	Geometric Verification	Total Time
Image Retrieval (region - 400x400)	~900 Surf features	1.95 TB	0.3s	10.9s	0.5s	0.4s	12.1s
Text Retrieval (block)	~100 words	5.5 GB	N/A	4.0s	N/A	N/A	4.0s
Image Retrieval (page - 2200x1700)	~6000 Surf Features	1.95 TB	2.6s	50.2s	1.8s	2.5s	57.1s
Text Retrieval (page)	~1000 words	5.5 GB	N/A	14.2s	N/A	N/A	14.2s

Table 10: Index sizes and average retrieval times for the image and text retrieval used in the experiments

As shown in Table 10, the vast majority of the time was spent on random disk seeks and reads. Hardware and software engineering improvements such as using solid state drives or adding more hard drives would likely greatly speed up this approach. Our image retrieval approach is still a magnitude slower than text retrieval algorithms, but results suggest that image retrieval is still usable because not all document images require image indexing and not all users require image queries.

2.5: Conclusion

2.5.1: Summary

To the best of our knowledge, this study is the first to conduct a large scale comparison to determine whether image retrieval can satisfy a user's information needs on a large real world dataset by scaling a segmentation free image retrieval algorithm to a 7 million document image dataset. In many cases when there are handwritten words, rare languages, obscure fonts, or noisy images where OCR is likely to fail, document image retrieval may be the only viable option. The results of this study are significant in showing that current image retrieval algorithms can be used to satisfy a user’s information need for general topic based queries on large heterogeneous datasets. The retrieval results on text, when combined with logos, performs at the state of the art level for the Tobacco 800 dataset.

As a baseline, our technique was compared to the retrieval of text obtained through OCR. Traditionally, this has been the most common approach for accessing document image collections. While the goal is to show that the combination of image retrieval and text retrieval would outperform text retrieval in general, it appears that on average text retrieval alone is still superior for the English text in the tobacco corpus. However, the image retrieval significantly outperformed text retrieval on a subset of the queries, and the combined image and text retrieval improved substantially more. This suggests that while the image retrieval algorithm is not needed in all cases, there exists a class of user topics and document images for which image retrieval is beneficial. Future research is required to identify the set of topics or use cases for which image retrieval technology can be the most useful. This is a variant of the query difficulty problem, which in general is known to be hard. In this case, however, we have evidence from OCR scores and word counts that could serve as useful features for query performance prediction [72]. The results also indicate that the relevance judgments from the TREC Legal dataset are biased towards the capabilities of text retrieval systems, and suggest future experiments in multimodal retrieval should try to include retrieval results from a larger variety of technologies in order to better support future use of the resulting collections.

2.5.2: Future Work

2.5.2.1: New Collections

While the approach presented in this chapter demonstrates the utility of content based image retrieval for general topic based user queries the dataset is no longer being actively worked on by a large community limiting the utility for the greater academic community of supporting such a system. There are several more recent collections being brought online such as “Franklin” [73], which is currently providing a growing collection of 700,000 scanned document images from the FDR presidential library online. These documents are actively being utilized by historians and the goal of the research would be to build a baseline system using OCR and document image retrieval to allow researchers to go through the data and work with the community to enable newer retrieval techniques to allow the historians to more efficiently work with these types of collections.

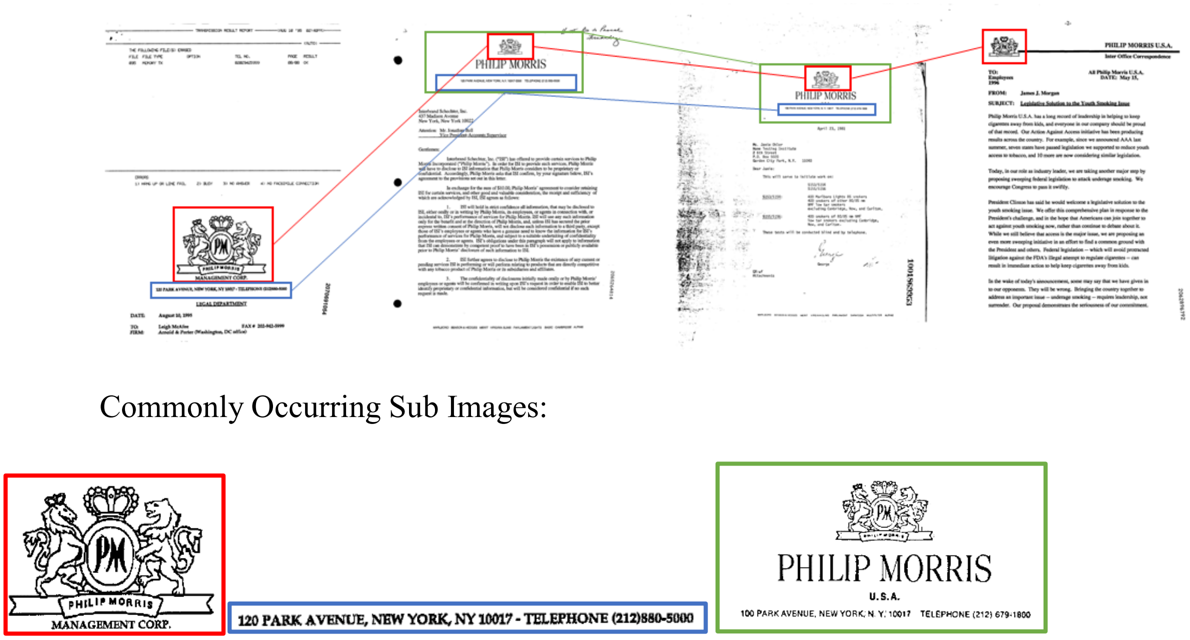
2.5.2.2: Improving User Queries Using Repeating Sub-Images

A second challenge for any image query-by-example system is how to generate the actual queries. In the evaluations presented in this chapter, it was assumed that sub-images were given or that the entire image was selected for relevance feedback by the user. In the cases where users were asked to select relevant sub-images from the results and to resubmit them as queries, they appeared to struggle with the task. Another interesting problem would be to look for visually repeating patterns or sub-images that occur in the Top N document images returned by an initial text query, and suggest them to a user. The hope is that query expansion using these suggested sub-images would more likely result in more relevant document images being returned than have users attempt to do this manually on a smaller subset of results. Examples of results from searching for the test “Philip Morris” from the tobacco collection are shown in Figure 21.

There are two possible ways in which we hope to find these reoccurring sub-images. A high level view of our first approach would be to load a smaller index containing these the SURF features from these top N documents into memory and use the existing matching and geometric verification framework to locate these sub-regions with high precision and efficiency. This would create a similarity graph between the N documents as visualized in Figure 21, where the nodes are sub-regions and the edges connect similar sub regions with high precision. To create a ranked order of sub-images for use in query expansion, we would use the minimum spanning tree of this graph to suppress commonly occurring sub regions and each node would be ranked by its degree. A second approach using segmentation of the images could also be explored. An approach, such as Voronoi segmentation [37], would split each document into M segments creating a total of N*M sub regions. These sub regions could then be clustered and the images closest to cluster centers, ranked by the cluster size, would be used for query expansion.

Query: “Philip Morris”

Results:


PHILIP MORRIS
PHILIP MORRIS
PHILIP MORRIS U.S.A.
Inter Office Correspondence
PHILIP MORRIS
MANAGEMENT CORP.
120 PARK AVENUE, NEW YORK, NY 10017 - TELEPHONE (212)880-5000
PHILIP MORRIS
U.S.A.
100 PARK AVENUE, NEW YORK, N. Y. 10017 TELEPHONE (212) 679-1800

Commonly Occurring Sub Images:

Figure 21: Example of learning relevant sub-images to an initial text query

2.5.2.3: Improving Retrieval Accuracy and Speed

There is also room for improvement in features and indexing used to create the segmentation-free document image retrieval system we developed. Currently a large number of interest points (~7000) are generated so that even very small sub-images could be accurately matched on the page. But not all features are useful when OCR is present. Even without OCR, we may find that similar retrieval accuracy could be achieved with fewer and larger interest points. While SURF has been shown to work well for binary images in a relatively low dimension feature vector, it would also be worthwhile to research better feature descriptors that are specifically designed to exploit the properties of document images. Finally, better hashing methods could be researched that achieve better precision and recall, while being less expensive both computationally and on disk.

Chapter 3: VisualDiff: Verification and Change Detection for Document Images

This chapter presents work on document verification and change detection. The goal of document verification is to provide a Boolean decision as to whether two document images contain identical content and layout or if changes are present. There are two main challenges associated with document verification. First the documents, even if the content is identical, could have significantly different pixel values due to changes from the camera or scanner capture of the document. Second, it becomes increasingly difficult to distinguish between noise and genuine content in cases due to poor binarization or degradation of the physical document.

Assuming changes are detected, the goal of document change detection is to determine precisely what and where changes have occurred given two similar document images. There are many reasons changes can occur between two document images with varying difficulty associated in detecting them. The first and easiest change to detect involves the addition of content without effecting the position or appearance of existing material on the page. This includes changes such as filling an existing form, stamping or signing a document, or appending data to the end of the document. A second, but more challenging case involves the addition, deletion, or modification of content into a structured document. In a best case scenario this only involves detecting a single translation vector for all existing material such as a paragraph being moved down. However, in the case where a single word is inserted into a body of text, this can cause cascading changes to each subsequent text line. Solving the correspondence problem to only detect the small portion where changes have occurred between the two documents can be further compounded by noise, complex layouts and word spacing. A similar issue comes up with the third type of change, which occurs when the content is identical, but the layout, style or formatting changes. This can include cases where a word is bolded or the format of the page is converted from one column to two columns.

For our work, we constrain the space of changes to the addition or deletion of content, which we assume is the most relevant to commercial applications. When designing our approach, we also assume missing a change can have more severe ramifications than false detections of regions containing changes.

3.1: Related Work

The problem of document verification can be viewed as a variation of early research in duplicate document detection. The goal was to develop approaches to reduce the replication of identical documents present in large databases. Initially, the focus was on imaging variations caused by multiple copies and general degradation of the physical instances, as opposed to any intentional markup. Since most of this work was done before mobile scanning devices gained popularity, most of the approaches are not robust to 3D pose change. We also note the duplicate document detection is distinct from near-duplicate detection, which is often used for retrieval, but is unsuitable for detecting if two images are indeed identical.

The most common first pass used by many researchers is a simple pixel difference for nearly identical documents. However, this is not invariant to common changes such as skew, scale, rotation, or even intensity differences, so researchers have typically used feature-based approaches. Doermann et. al. [74] creates a signature using simple properties of characters extracted from text lines, and is able to detect 93% of degraded documents. Hull [75] imposes a grid upon the image and extracts feature vectors based on pass codes to determine if corresponding grid locations of the images are identical. 95% of matches are correctly identified with most errors caused by skew and scale changes. Lopresti [76] uses the edit distance and vector space model on OCR'd text extracted from images to determine their similarity and if they are indeed duplicates. However, his technique assumes reasonable quality OCR can be obtained from the image. Most recently in [77], Jiang et. al. use connected component features to develop a hashing function that detects changes in similar documents. They test on 120 images scanned at 600 DPI with artificially placed modifications and obtain a 100% detect rate, with a 2.5% false acceptance rate.

Since most of these image based techniques for duplicate detection were developed prior to the popularity of camera capture of document images they are not robust to changes in scale, rotation, and perspective change. Furthermore, the use of connected components can lead to poor performance in the cases of touching components or broken characters and assumes a binarization step that may not be necessary. OCR is also a poor choice because even with 99% character accuracy there will still be several character errors present in the page and poor page segmentation of complex documents common in heterogeneous collections can lead to poor OCR accuracy. All of the techniques are also primarily designed for documents containing text and it appears they would fail when presented with graphical objects, table or graphical objects present in the page.

Change detection is a common problem for both text and multimedia content. Early work on change detection in text was based on the longest common subsequence problem. Meyers [78] presented an efficient solution to this, which is still used as the implementation in the popular UNIX program *diff*. Since that time many variations have been created that are able to find changes between two text documents at the character, word or sentence level. Change detection has also been an important research area for images and videos. Here changes between similar images or frames are used to characterize an optical flow, which can be used to describe the motion of the camera or structure of objects within the images [79]. Change detection has also been applied in the document image domain for the purposes of document authentication. The authors of [80] designed a verification system using error correction codes to detect pixel level changes and verify content integrity. Their experiments show they are able to detect 97% of pixel changes, but the weakness is that the document must physically contain four markers placed by the system for localization and a barcode with the error correction codes. However, this approach does not account for the layout or structure of a document as content is modified resulting in many more changes being detected than is necessary, potentially requiring a user to examine the whole page.

Three recent studies have explored applications of change detection for document images using OCR'd text. Clough et. al. [81] examined the problem of text reuse to examine how news articles changed as they were reprinted. The changes detected from text provided evidence of biases, writing styles, or vocabulary limitations of the news organizations. Sayeed et. al. [82] studied methods to determine whether modifications in document images of structured contracts were compliant using document similarity measures. Alexander Rush introduced an automated approach for redaction analysis, which analyzes different versions of declassified documents containing redacted passages in order to recover the redacted text. His approach relies on finding and aligning related document images using OCR'd text and then detecting redacted passages using additions or deletions found in the alignment [83]. These approaches are dependent on the quality of the OCR'd text and we believe that the image-based change detection approaches presented in section 3.3 could be extended to improve these applications in cases where the OCR quality is poor or the changes of importance are not obtainable through OCR.

3.2: Document Image Verification

We propose a more robust solution to document verification that can cope with common image transformations and does not require the binarization or segmentation steps common with connected component approaches. The main contributions of our approach, outlined in Figure 22, are to first align two images into the same coordinate space by finding their homography and then to find pixel level changes using a dense SIFT correspondence.

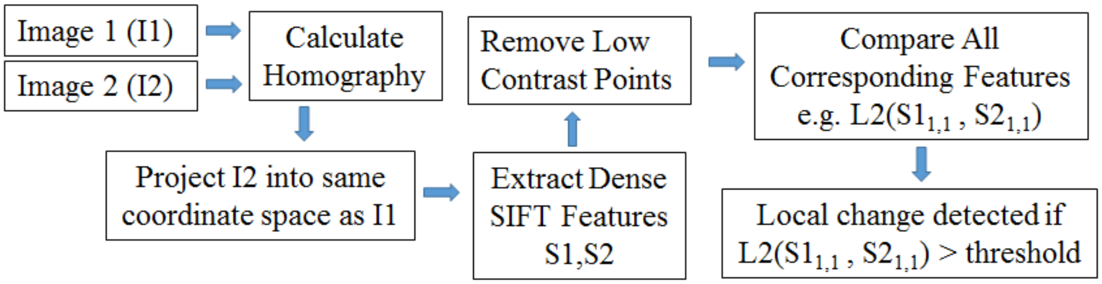


Image 1 (I1)
Image 2 (I2)
Calculate Homography
Project I2 into same coordinate space as I1
Extract Dense SIFT Features S1,S2
Remove Low Contrast Points
Compare All Corresponding Features e.g. L2($S1_{1,1}$, $S2_{1,1}$)
Local change detected if L2($S1_{1,1}$, $S2_{1,1}$) > threshold

Figure 22: Document Verification Approach

Our first step for document verification aligns two images to remove changes caused by rotation or translation of the camera or scanner. Document images can be approximated by a planar surface in three dimensions. Thus a homography matrix can be used to describe a transformation to project the pixels of one image onto the other. Processes to obtain this matrix are well known in computer vision where a set of local features are extracted from an image and then matching descriptors are used to find the homography matrix [84]. Recent research in document image retrieval has shown SIFT [62] to perform well for detecting local correspondences in document images and is hence used as the local descriptor. Even though, SURF was superior for binary images, SIFT appeared to work better when directly comparing grayscale documents to binarized images. We use RANSAC to remove outlier matches during the calculation of the homography matrix. This process is illustrated in Figure 23.

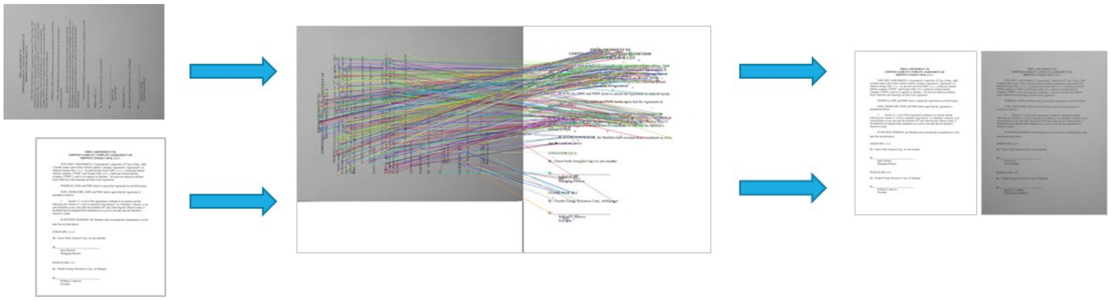


Figure 23: (a) Alignment using SIFT between two similar images.

Once the two images were aligned, a set of dense SIFT descriptors [85] was extracted from even intervals in both images. SIFT was chosen since it has been shown to be invariant to small changes in lighting, blur, skew and out-of-plane rotations [62]. Additionally, it can be used for binary and grayscale images without requiring segmentation unlike previous descriptors based on connected components, which were largely developed for binary images. One of the reasons SIFT works so well is that it is essentially capturing the local edge information, which is similar to what would be extracted from the contour of connected components. Descriptors from corresponding positions in both images are compared and a change is said to be present at a given location in the image if the Euclidean distance between any two SIFT feature vectors (S1,S2) at position x,y in images I1 and I2 falls above a threshold *t* as shown in Equation (11). The final metric used for determining if a change was present, $Change(I1, I2)$, is the sum of local changes when comparing both images to each other as shown in Equation (12).

$$Diff(I1, I2) = \sum_{y=1}^{Height} \sum_{x=1}^{Width} \begin{cases} 0 \; if \; Contrast(I1_{x,y}) == 0 \\ 0 \; if \; L2Dist(S1_{x,y}, S2_{x,y}) < t \\ 1 \; if \; L2Dist(S1_{x,y}, S2_{x,y}) > t \end{cases} \qquad (11)$$

$$Change(I1, I2) = Diff(I1, I2) + Diff(I2, I1) \qquad (12)$$

Figure 23 shows an example of changes detected between two similar document images following alignment and the dense feature comparison. In practice however, SIFT proved to be unstable in regions that predominantly had white or black space, where small amounts of noise or illumination changes would dominate the gradients used to create SIFT. Thus, areas of low contrast were not used when performing the dense feature comparison as shown in Equation (11). Using a simple threshold for detecting changes worked well because even small one-character differences cause large changes in the underlying SIFT descriptor between the two images, while regions with identical content were very close even in Euclidean space so long as the homography estimation was accurate.

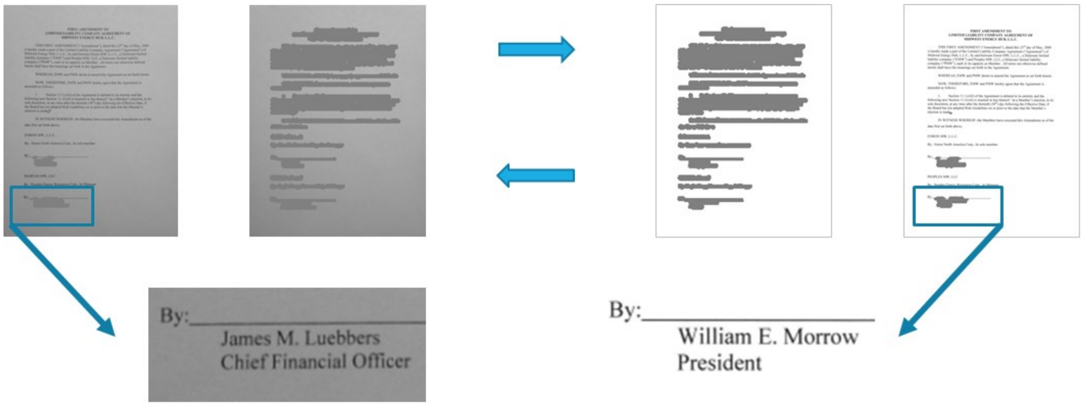


Figure 24: Dense SIFT feature comparison, excluding low contrast areas

SIFT proved to be robust to small pixel shifts, but larger shifts caused by curvature in the page when scanned using a camera phone, caused problems since the alignment assumed only linear changes were present. Rather than trying to model this curvature, it was sufficient to search other descriptors in a small neighborhood around the corresponding point in the aligned image and search for the minimum SIFT distance. This leads to the updated equations (13) and (14), where *w* is the width of the window search.

$$N(S1,S2) = \min\left(L2Dist\left(S1_{x,y}, S2_{x+w,y+w}\right), \ldots, L2Dist\left(S1_{x,y}, S2_{x-w,y-w}\right)\right) \tag{13}$$

$$Diff(I1,I2) = \sum_{y=1}^{Height} \sum_{x=1}^{Width} \begin{cases} 0\ if\ Contrast\left(I1_{x,y}\right) == 0 \\ 0\ if\ N(S1,S2) < t \\ 1\ if\ N(S1,S2) > t \end{cases} \tag{14}$$

The misalignments in two overlaid images and the local neighborhood search is illustrated in Figure 25. In order to speed up feature comparison, we found the dense SIFT comparison can be done using images resized to 100 DPI rather than full resolution and using a dense grid sampled at every other pixel with no loss in accuracy.

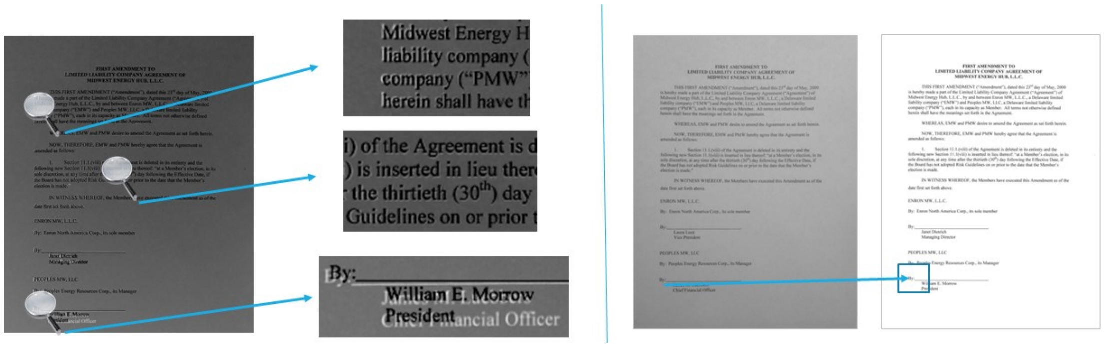

Midwest Energy H
liability company (
company ("PMW"
herein shall have th
i) of the Agreement is d
) is inserted in lieu ther
the thirtieth (30th) day
Guidelines on or prior t
By:
William E. Morrow
President

Figure 25: Misalignments caused by small page warping (left). Local neighborhood search (right)

3.3: Document Image Change Detection

Although it is useful to detect if changes have occurred in a document image, many users could also want to determine precisely what has changed without having to manually scan documents character by character. Three approaches are examined for change detection in document images as shown in Figure 26. The goal of these approaches is to detect locations on the document images that have content changes, while minimizing false positives. Typically this is accomplished by performing OCR on the documents and using text difference utilities like UNIX *diff* to identify the changes. Hence, the first method, used as a baseline, performs "diff" on the OCR text extracted from the images; using the longest common subsequence (LCS) algorithm to identify changes. If there are OCR errors using a text-based diff will lead to many false positives, so the second approach extends traditional techniques for finding the LCS in text to images. It performs a "diff" using LCS on SIFT features extracted from line images. While the previous two techniques rely on page and line segmentation, the third technique performs a segmentation free alignment of corresponding SIFT features on the page to identify changes.

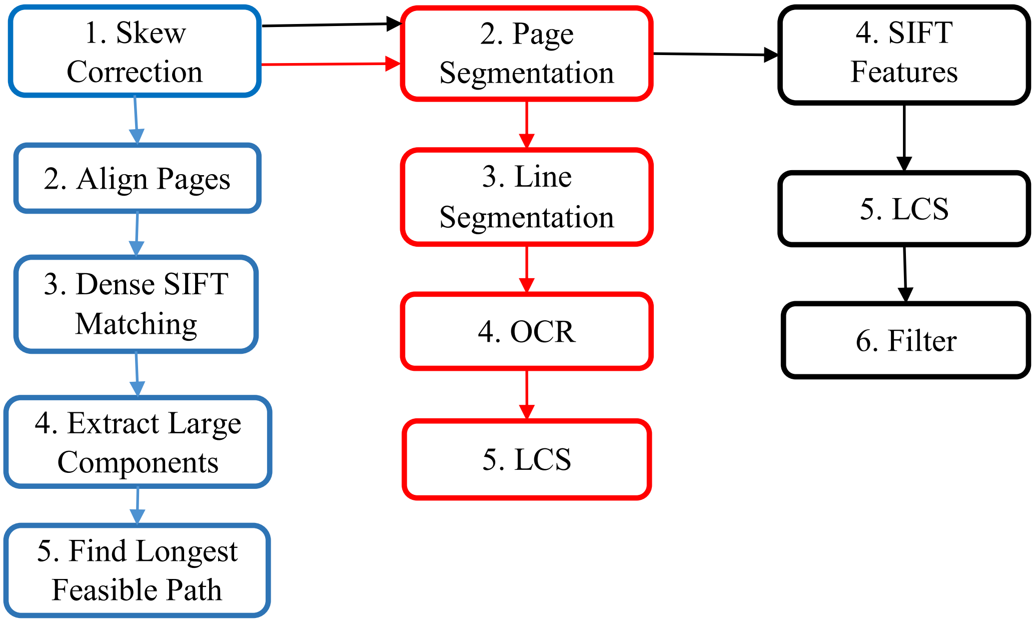


Figure 26: Outline of the three document Image change detection techniques. VisualDiff++ (BLUE), OCR Diff (RED), and SIFT Diff (BLACK)

3.3.1: Longest Common Subsequence

Given two sequences $X_{1,2..m}$ and $Y_{1,2..n}$ the goal of the longest common subsequence (LCS) algorithm is to find the longest ordered subsets shared by both X and Y. The opposite of this problem is to find the shortest set of differences, which is exactly what is needed for change detection. Hence, change detection for traditional text can be thought of as an extension to the LCS problem, where text that is not shared between two documents in the LCS is considered a deletion if it only exists in the original document and an addition if it only exists in the new document. The LCS can be found from the following recursive function given two sequences (*X*, *Y*):

$$LCS\ (X_i, Y_j)\ = \begin{cases} 0\,, \quad if\ i = 0\ or\ j = 0 \\ LCS(X_{i-1}, Y_{j-1})\ +\ 1, if\ X_i = Y_j \\ max(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)), if\ X_i \neq Y_j \end{cases} \qquad (15)$$

To obtain the LCS, one can keep all elements where $x_i = y_j$ for the longest LCS. Deletions are defined by the set of elements in X, which are not part of the LCS and likewise additions can be found from the set of elements in Y that are not in the LCS. The complexity for this naïve approach is O (n*m), when only two sequences are involved. However, Myers extended this in [78] by proposing a greedy heuristic where the problem could be solved in O (n*d) where d is the number of edits between the two versions. For efficiency reasons, *diff* has traditionally only computed the LCS and edit distances for lines of text rather than at the character or word level. In many cases this is beneficial to allowing the user to understand the context of the change. However, the addition of one word can also shift subsequent words into different lines leading to newer "*diff*" applications such as the one used by Microsoft Word, which work at word level.

3.3.2: OCR Diff

As a baseline approach, a similar method can be employed with document images by performing the "diff" on the OCR'd text extracted from the image using the LCS algorithm. OCR text can have minor errors so instead of just using string equality, the Levenshtein edit distance is used to allow for a string equality operator tolerant to minor errors. Two words are said to be equal if the edit distance is less than a threshold, T_1. The full process used for obtaining the OCR and performing the change detection is shown in Figure 26. The pages were deskewed using OCROpus [29] and line and page segmentation were performed by the OCR Engine. We initially tried two open source OCR engines, but found that the character error rate and errors from segmentation were too high to make a reasonable baseline and instead chose to use OMNIPage, a commercial OCR engine. It produces a PDF file from which each word and text line along with the bounding box of their locations on the document image can be extracted.

Using OCR for change detection suffers from a number of drawbacks. First, the accuracy of the approach is heavily dependent on successful preprocessing to extract good lines and page segmentation. Poor segmentation could result in entire portions of the image being unevaluated for changes. Second, OCR can often have many small errors leading to false positives and the error rate can vary widely between well-studied, simpler scripts such as Latin and more complicated scripts such as Arabic and Devanagari. Finally OCR engines are built for machine print text, so graphics, logos, stamps, handwritten edits, or signatures are not guaranteed to be processed, potentially missing important changes.

3.3.3: SIFT Diff

LCS can be extended to work with sequences of features or images instead of text as long as there is an equality operator for any two elements. In order to extend the baseline approach in cases where OCR has high character error rates, SIFT features extracted from the line are used in lieu of OCR'd characters. An overview of the approach is summarized in Figure 26, where the deskewing and segmentation are identical to the previous approach. The lines were obtained from the OMNIPage PDFs and normalized to a fixed height of 32 pixels while maintaining the aspect ratio. Each of the lines is concatenated to form one long image and then SIFT descriptors are extracted at regular intervals on the horizontal center of the line image as shown in Figure 27. This is much more efficient than the dense extraction and comparison used in section 3.2.

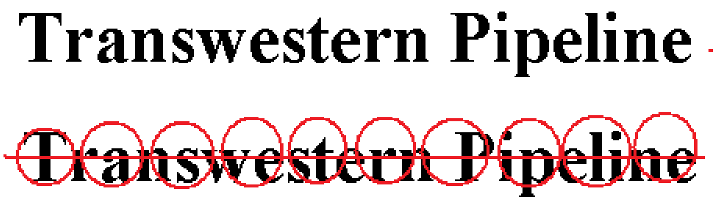


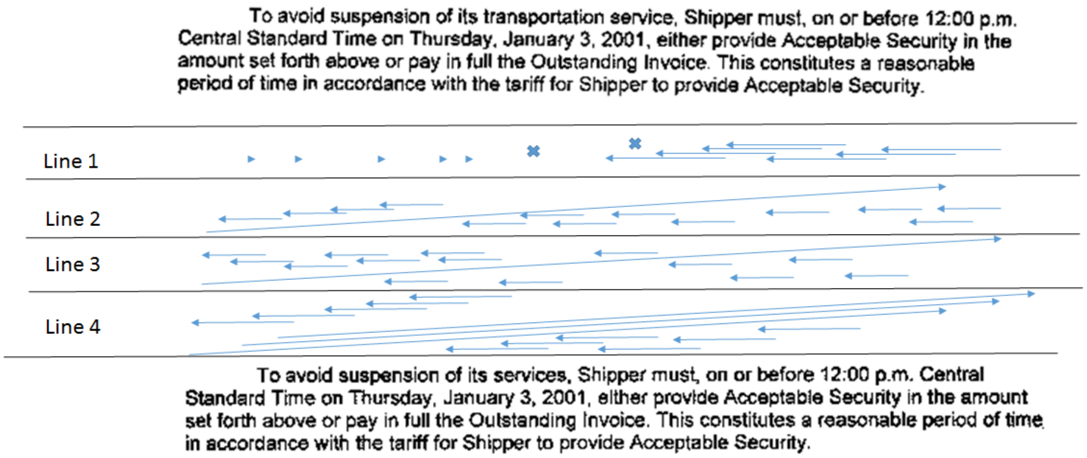
Figure 27: Line Image Feature extraction

In order to detect changes using LCS, two SIFT descriptors are said to be equal if their Euclidean distance is less than a threshold, T_2. Here the LCS is instead performed at the feature level, allowing the approach to find local changes at the character or word level. In addition to the legitimate content-based changes identified by LCS, false positives also occur due to misalignments caused by spacing differences between letters from the line concatenations and slight scale differences during the line normalization, which causes one line of text to be slightly larger than an identical corresponding line. In order to filter the majority of these false positives, in practice, we found it to be useful to only include changes which had at least N consecutive neighboring changes and ignore changes in low contrast areas that corresponded to large areas of white space in the line.

This approach shares a similar weakness with OCR in that it is heavily dependent on good segmentation. If the reading order of the automatically extracted page zone or lines changes between the two document images, this can significantly affect the performance of the approach. Worse still are cases where the segmentation fails, leaving portions of the page unevaluated for change detection. This motivates our new approach, which is segmentation free.

3.3.4: VisualDiff++: Segmentation Free Document Image Change Detection

When text is added or deleted from documents, the blocks of characters (words, sentences or paragraphs) on the page shift as defined by the page layout. If a word is added, the remaining content may shift to the right and in some cases cause a cascading series of shifts of varying sizes in subsequent lines in the paragraph. Similarly, if a paragraph is added to a single column document then the remaining text would just shift directly down. In more complex layouts, such as multi-columned documents, the text may shift to the upper right if the end of a column is reached in addition to shifting down or to the right. In each of these cases, as revisions are made, there are many blocks of text shifting throughout the page defined by different X, Y translations. This intuition, demonstrated in Figure 28, guides the VisualDiff++ approach.



To avoid suspension of its transportation service, Shipper must, on or before 12:00 p.m. Central Standard Time on Thursday, January 3, 2001, either provide Acceptable Security in the amount set forth above or pay in full the Outstanding Invoice. This constitutes a reasonable period of time in accordance with the tariff for Shipper to provide Acceptable Security.

To avoid suspension of its services, Shipper must, on or before 12:00 p.m. Central Standard Time on Thursday, January 3, 2001, either provide Acceptable Security in the amount set forth above or pay in full the Outstanding Invoice. This constitutes a reasonable period of time in accordance with the tariff for Shipper to provide Acceptable Security.

Figure 28: Shifts between words lines in two paragraphs where only one word is deleted

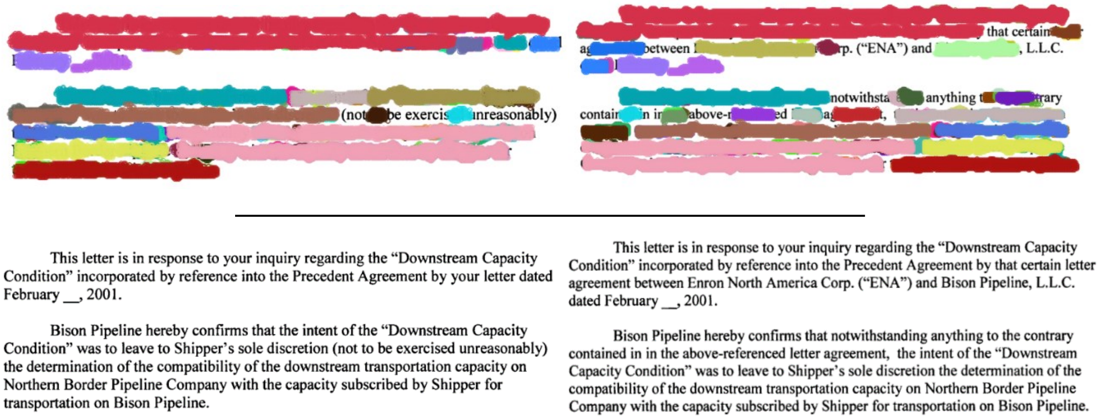
1: Finding Matching Blocks of Text

To simplify the problem, we assume that neither of the document images contains significant perspective or affine distortions. This is typically the case for images scanned on flatbed or autofeed scanners, but is also reasonable for mobile scanner applications such as CamScanner, which assists the user in finding the 4 corners of the page in order to warp the image back to a flat 2D plane. The images are first deskewed using OCROpus to ensure that both images are upright. Next, one image is projected onto the other using a Transformation matrix (H), which is estimated by performing RANSAC on SIFT keypoint matches between the two images. To improve the stability of the estimation, H is constrained to exclude affine and perspective changes:

$$H = \begin{bmatrix} s * \cos(\theta) & \mathrm{s} * \sin(\theta) & T_x \\ -\mathrm{s} * \sin(\theta) & s * \cos(\theta) & T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Once the two images are aligned and upright, dense SIFT descriptors are extracted from even intervals on both pages. Descriptors in regions with low contrast such as whitespace are discarded. In order to identify matching blocks of text between the two pages, the L2 distance from each of the dense SIFT descriptors on one page is compared to the descriptors on the second page and the matches with a distance less than a threshold of T_3 are retained. To speed up this computation a forest of KD-trees [86] is used to perform efficient approximate nearest neighbor search.

Since there are no scale or rotation differences between the two images once they are aligned, the projection of a point from one image onto the other is given by an X, Y translation. A grid is formed by partitioning the X, Y translation space into 4x4 pixel blocks. Matches are binned into the compartment defined by their translation as well as all 8 adjacent bins in order to efficiently find potential blocks of text that shifted the same amount. Matching pairs of connected components are formed by merging neighboring keypoints within the same bin with a mass greater than 20 keypoints. This helps to filter small random matches and reduce the overall number of components for efficiency reasons. Once this is completed, connected component pairs that have over 90% keypoint overlap in both images are also merged to reduce the number of matching components again for efficiency. An example is shown in Figure 29, where the connected component pairs found are paragraphs, lines, words, and even partial words.


(not be exercis unreasonably)
that certain
between
p. ("ENA") and
, L.L.C.
notwithsta
anything
rary
contai
n i
above-
ed
ag

This letter is in response to your inquiry regarding the "Downstream Capacity Condition" incorporated by reference into the Precedent Agreement by your letter dated February __, 2001.

Bison Pipeline hereby confirms that the intent of the "Downstream Capacity Condition" was to leave to Shipper's sole discretion (not to be exercised unreasonably) the determination of the compatibility of the downstream transportation capacity on Northern Border Pipeline Company with the capacity subscribed by Shipper for transportation on Bison Pipeline.

This letter is in response to your inquiry regarding the "Downstream Capacity Condition" incorporated by reference into the Precedent Agreement by that certain letter agreement between Enron North America Corp. ("ENA") and Bison Pipeline, L.L.C. dated February __, 2001.

Bison Pipeline hereby confirms that notwithstanding anything to the contrary contained in in the above-referenced letter agreement, the intent of the "Downstream Capacity Condition" was to leave to Shipper's sole discretion the determination of the compatibility of the downstream transportation capacity on Northern Border Pipeline Company with the capacity subscribed by Shipper for transportation on Bison Pipeline.

Figure 29: Above: Matching blocks of text (connected component pairs) found in two similar images. Below: The actual text from the two document images.

2: Detecting changes by finding the longest feasible path

In the previous two approaches, the reading order is provided by the page and line segmentation so a straight forward implementation using LCS is possible. However, in this segmentation-free change detection scenario there is only a list of matching connected component pairs, which represent text blocks in common between the two images. Our goal is to find an ordered list or path of matching text blocks, with the largest combined mass, that creates a feasible reading order in both images. In order to enforce the reading order, a directed graph is created by drawing an edge from one connected component pair (CCP1) to another connected component pair (CCP2) if the following constraints are met:

1. Neither connected components in CCP1 and CCP2 have more than 10% keypoint overlap with each other in their respective images.
2. Both connected components in CCP2 are either to the right or below the connected components in CCP1 in their respective images.

The intuition behind the first rule is to only allow a path in the graph to cover a text region once. The intuition behind the second rule is that the reading order goes from left to right until the end of the line is reached and then down to the next line. For multi column documents it is also possible to go up and to the right. However what is excluded from the rule and is not possible is for the next word to go toward the upper left between 90 and 180 degrees.

Finding the path in the graph with the largest combined connected component pair mass is analogous to the longest path problem in directed graphs, which is known to be NP-hard in general cases. In order to make the computation tractable we use a beam search [87], with a heuristic based on choosing the next node based on the largest potential path size if the next node were to be added. In order to enforce the reading order, potential nodes that are not directly connected to all the vertices in the current path are pruned from the search and are not included in the potential path size. The beam search is not guaranteed to find the largest path, but in practice is generally close to the ideal path sometimes excluding a few of the smaller components. Once the longest path is found, the connected components in the path are dilated by 5 pixels to cover any small gaps that occurred around the boundaries. The set of original dense keypoints that did not overlap with the dilated connect components in the largest path are then marked as changes.

3.4: Experiments

3.4.1: Datasets

3.4.1.1: The Enron Revisions Dataset

The problem of document change detection can be studied from the viewpoint of tracing a revision history. Given a set of revisions in a document, one can get a before and after snapshot of a document to create a ground truth dataset for change detection. An easy source of these types of revisions can be found in Microsoft Word documents containing track changes. In order to obtain a real world set of revisions, we looked to the Enron document collection [2], which contains about 500,000 attachments, including over 180,000 Microsoft Word documents. We wrote a script to extract documents containing modifications from Microsoft Word's track changes feature. 150 one-page documents were selected from this collection, each containing between 4 and 41 additions or deletions of text. These text modifications ranged from one character to entire paragraphs. Changes including formatting and font changes were not retained for this study. The documents themselves contain letters, memos, and contracts. Since this dataset is a subset of the Enron collection, we named the dataset the "Enron Revisions" collection. Examples of these documents can be seen in Figure 30.

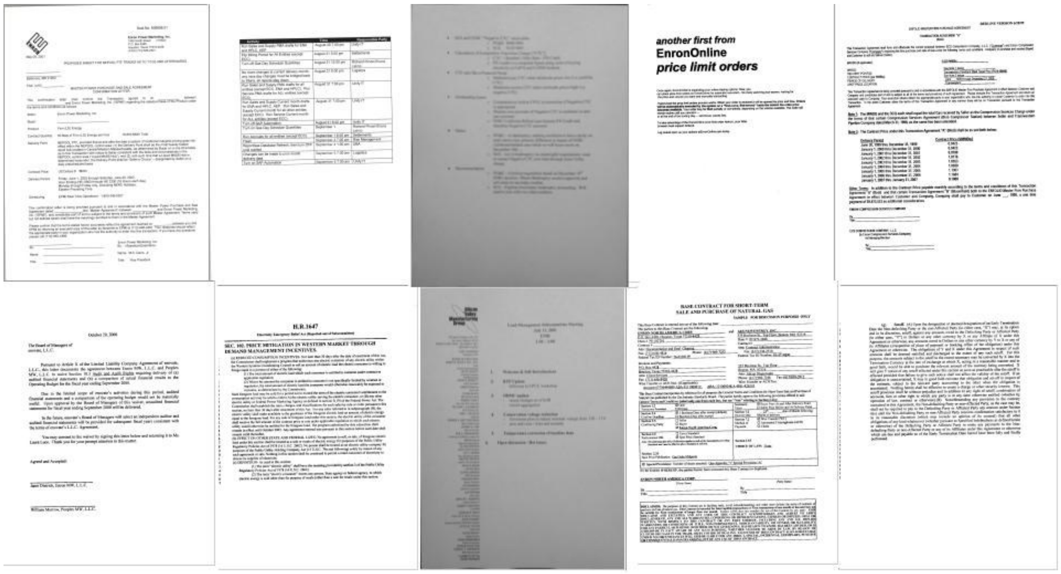

another first from
EnronOnline
price limit orders
H.R.1647
BASE CONTRACT FOR SHORT-TERM
SALE AND PURCHASE OF NATURAL GAS

Figure 30: Images from the Enron Revisions dataset

A hardcopy of the before and after snapshot of each document, (created by accepting or rejecting the track changes), was printed. To study the effectiveness of the document verification and change detection to common forms of variation, 11 document images were created for each of the 300 (2 x 150) pages. Six variations were created from an autofeed scanner including images scanned as: 100 DPI, 100 DPI binarized, 200 DPI, 200 DPI binarized, 300 DPI, and 300 DPI binarized. Many of the images had varying amounts of skew from the auto-feeder. The remaining variations were created from an iPad using CamScanner, a popular app for scanning on mobile devices that automatically crops the page and attempts to flatten the image and remove affine changes. The iPad camera created an image with resolution equivalent to ~200 DPI. From that image the following five variations were created: original, binarized, 2x2 motion blur, 4x4 motion blur, and 6x6 motion blur. All images taken from the iPad were at a 90 degree rotation, contained shadows and lighting changes, and slight out-of-plane rotations. In order to limit unintentional changes the iPad was stabilized while taking pictures and blur was added later using a motion blur kernel in order to simulate shaking of the hand. Motion blur beyond a 6x6 kernel made the image illegible and we assume a user would retake the picture at that point. Figure 31 contains image of the word “TRADES” under each of the 11 variations.

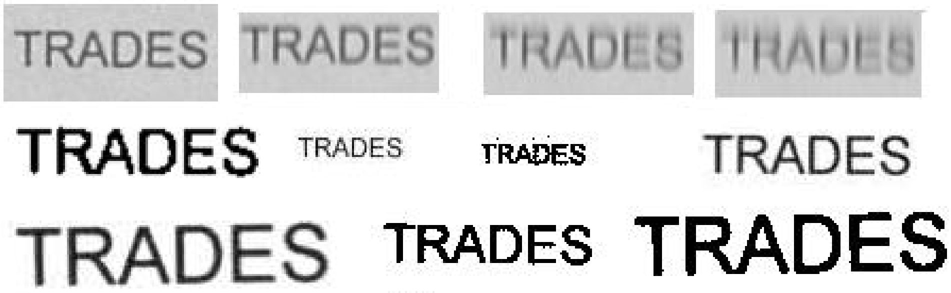


Figure 31: Example of distortion from the 11 scanning variations.

3.4.1.2: Tobacco Near Duplicate Dataset

While the Enron Revision dataset is well suited for studying document change detection in a controlled setting under varying conditions, it may not reflect the complexity of real world document image collections. These collections may be noisier and have more challenging page layouts. To address this concern, another dataset was created by finding near duplicate pairs containing content changes from the Tobacco collection [1], which contains binarized document images scanned at resolutions ranging from 100 – 300 DPI. More specifically, 100,000 document images from the collection were represented with a bag of SURF features and for each image the closest document, measured by cosine similarity, in the collection was retained. The top 10,000 pairs were then randomly sampled and the first 100 pairs were retained, discarding exact duplicates, completely unrelated document images, or pairs containing a document that was already utilized. As shown in Figure 32, these documents were more challenging because they contained handwriting, tables, graphics, and a variety of layouts. Each word, signature, and graphic was manually annotated as either being in common or changed between the two versions.

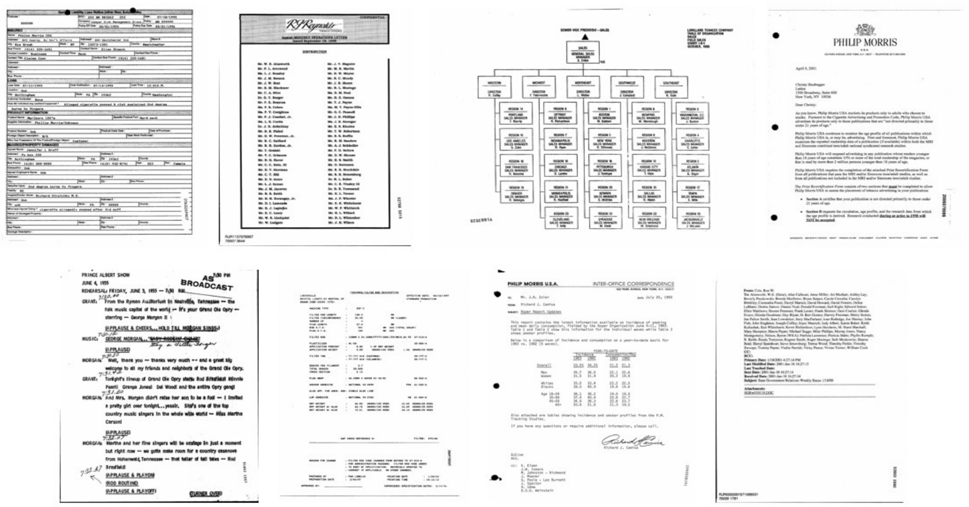


Figure 32: Images from the Tobacco Near Duplicate Dataset.

3.4.1.3: Experimental Setup

50 of the document pairs from the Enron Revisions dataset were set aside for parameter tuning and algorithm testing prior to running the experiments. The remaining 100 document pairs in the Enron dataset were used in the document verification and change detection experiments. The 100 document pairs from Tobacco NearDupe dataset were only used to evaluate the change detection approaches. For document verification, the window size (*w)* was set to 20 pixels and the SIFT L2 distance threshold (*t)* was set to 225. A simple threshold was sufficient for this problem because even small changes in content caused large discrepancies in the underlying SIFT descriptor, while identical content is very close in feature space as shown in Figure 33. For change detection, we set the number of consecutive changes needed for the SIFT Diff filter to N=2 and the size of the beam in VisualDiff++ to 20.

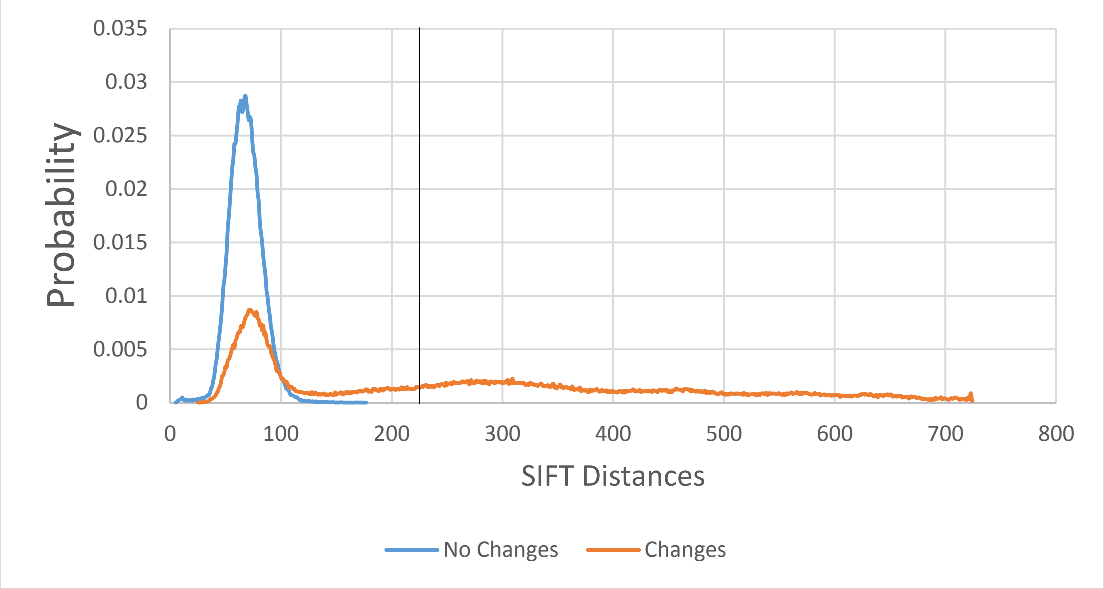


Figure 33: Probability of the distance of two corresponding SIFT features for document verification when a change is present versus no change is present. Threshold is shown as a black line.

3.4.2: Document Image Verification Results

In order to verify the effectiveness of the document verification procedure described in Section 3.2, each of the documents from the Enron Revisions dataset is compared to all before and after variants. As a baseline we use OCR extracted using OmniPage for each of the images. Even with the stronger OCR engine, the overall word error rate was 15%, with error rates as high as 70% in the blurred mobile documents. All non-alphanumeric characters and extraneous spaces were removed from the OCR output to try and correct simple OCR errors. We use the Levenshtein edit distance measure, which is similar to the distance measure presented in [76], to detect document image duplicates from OCR.

The ROC curve created by varying the edit distance threshold for OCR and number of miss-matching dense features is shown in Figure 34. The results show that the SIFT based verification procedure significantly outperforms OCR for duplicate

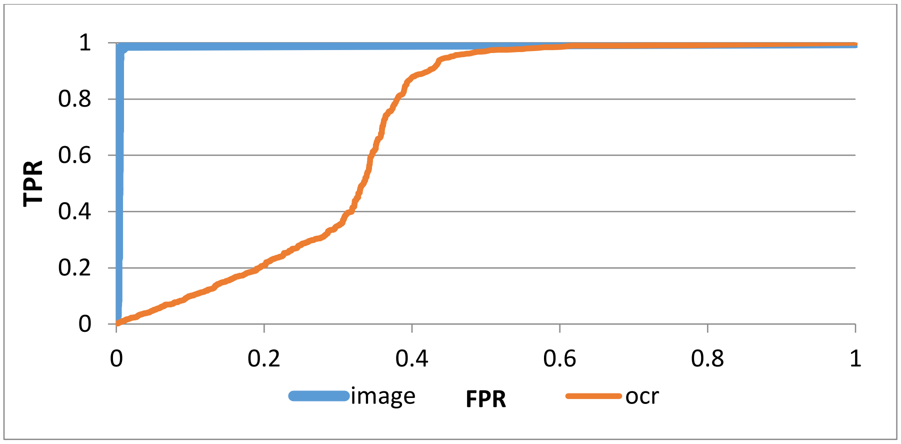


Figure 34: ROC curve for SIFT and OCR based document verification

detection on this dataset. The majority of errors could be traced to large errors in the OCR on the iPad images, especially the blurred ones. There were only a few false positives from the SIFT detector with zero false negatives. Over 99% of the identical document variants had zero dense SIFT differences, meaning the entire document was considered to match perfectly. Figure 35 displays two images that cause most of the false positives from the SIFT based approach due to severe blurring and binarization of darkly colored regions.

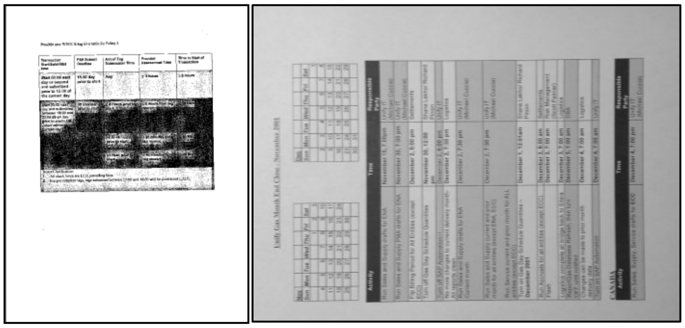


Figure 35. Images that cause false positives for document verification due to binarization and blurring

3.4.3: Document Image Change Detection Results

Both of the datasets are annotated at the word level, with bounding boxes indicating the regions of the image belonging to a given word or graphic. Each of the three change detection algorithms returns a set of X, Y coordinates on the image where changes corresponding to differences in the lines or SIFT keypoints occur. A change is said to be present if the algorithm under evaluation reports even a single change anywhere within the bounding box.

A ROC curve is generated by plotting the true positive rate against the false positive rate for each approach by varying the Levenshtein Edit Distance for OCR'd text or Euclidean Distance threshold for SIFT features used for equality. In this context a positive is the detection of a change and negative is the absence of change. The Area Under the ROC Curve (AUC) is also reported along with the True Positive Rate (TPR or Recall) and False Positive Rate (FPR) at the point on the operating curve with the highest recall for the baseline (OCR Diff) approach (string equality or Levenshtein Edit Distance=0).

$$True\ Postive\ Rate\ = \frac{true\ positives}{true\ positives\ + false\ negatives} \tag{17}$$

$$False\ Positive\ Rate\ = \frac{false\ positives}{true\ negatives + false\ positives} \tag{18}$$

3.4.3.1: Enron Revisions Results

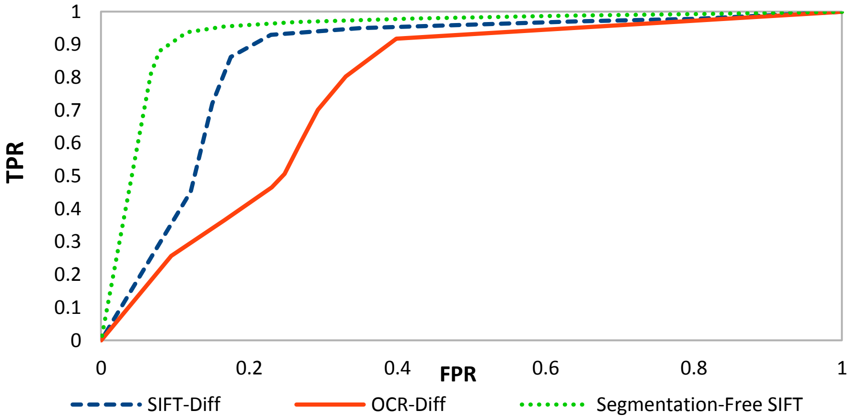


Figure 36: ROC curve on the Enron Revisions dataset

All 11 versions from the before and after snapshot are evaluated against each other for change detection. The results below show the average across the versions and 100 samples. The segmentation free method is clearly the best on this dataset. Compared to OCR, it has a 75% reduction in error for FPR and a 66% reduction in error for the ROC AUC as shown in Table 11. The separation is also clear on the ROC curve in Figure 36. The average OCR error rate is fairly high on this dataset due to the low resolution of some documents as well as mobile scanned documents containing blur. Hence the SIFT Diff method excels over OCR in this setting. Line and Page segmentation errors were not a major concern since only seven of the documents had difficult layouts with more than a single column, though the segmentation did error on some of the mobile scanned documents with significant blurring.

Change Detection Results	TPR	FPR	AUC
OCR Diff	92.4%	39.9%	0.752
SIFT Diff	92.4%	22.7%	0.838
VisualDiff++	92.4%	11.3%	0.913

Table 11: Change detection results on the Enron Revisions dataset

3.4.3.2: Tobacco NearDupe Results

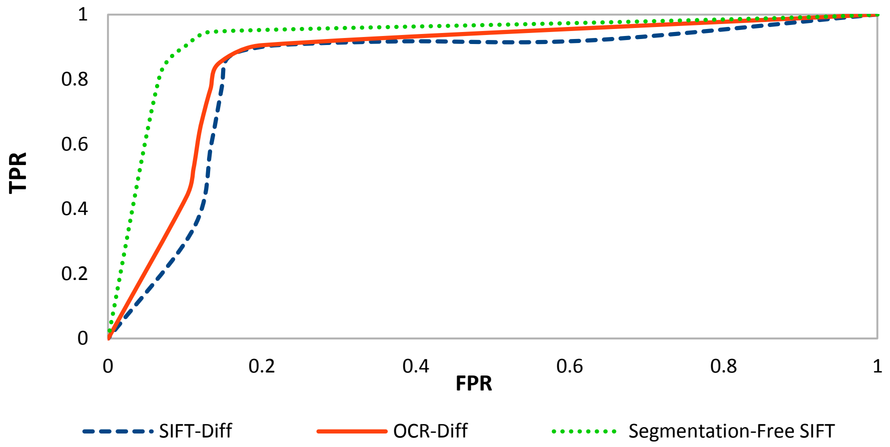


Figure 37: ROC curve on the Tobacco NearDupe dataset

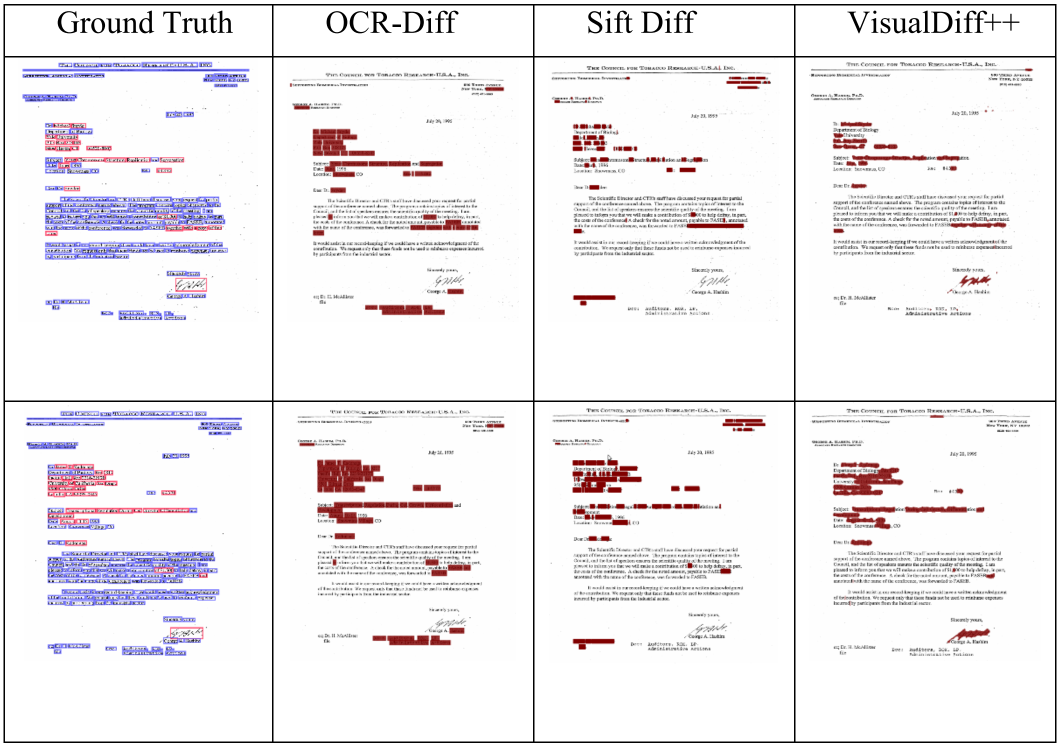


Figure 38: Changes detected highlighted in red and shared text in blue. Note that VisualDiff++ is much closer to the ground truth in Column 1.

Each of the three approaches were evaluated on the 100 pages in the Tobacco NearDupe Dataset with the results shown in Figure 37 and Table 12. VisualDiff++ outperformed both other methods, reducing the error rate by 50%. This was due to the algorithm's robustness to the more complex layouts when segmentation failed to correctly identify all the text regions in the page or the OCR engine failed to process graphics and handwriting. An example of this improvement is shown in Figure 38. OCR Diff and SIFT Diff worked similarly, both being equally limited by segmentation errors. The OCR based *diff* performed slightly better due to the OCR engine being more robust to noise. The character error rate was much lower than the Enron revisions dataset. This was likely due to the pages being scanned and binarized in typical settings as opposed to the lighting variation, blur, and extreme low resolution present in the Enron dataset.

Change Detection Results	TPR	FPR	AUC
OCR Diff	90.0%	19.5%	0.861
SIFT Diff	90.0%	19.6%	0.833
VisualDiff++	90.0%	9.5%	0.921

Table 12: TPR, FPR, and Area under the ROC curve reported on the Tobacco NearDupe dataset.

3.5: Conclusion

3.5.1: Summary

The work presented in Chapter 3 details a generalized approach for document verification, which is shown to be robust against common transformations that come from traditional scanning as well as camera capture, including binarization, resolutions changes, motion blur, and intensity changes. We also present two approaches for change detection in the image domain, limited to the addition or deletion of content. Results demonstrate that our segmentation free change detection approach results in fewer false positives than when using OCR.

3.5.2: Future Work

3.5.2.1: Beyond Additions and Deletions

One may also have to detect additional changes beyond the additions, deletion, and substitution of content. For example, changes in font style, size or color may indicate important sections in the document. Graphics, handwritten annotations, or tables may be inserted or shifted within the document. The page layout could be significantly changed by increasing the number of columns or altering the margins without any true content changes. The hope is to create a system that emulates track changes in Microsoft Word to list the locations and types of changes between the two documents, while presenting this information in a concise manner.

3.5.2.2: Learning from Changes

Correspondences between near duplicate documents also presents an opportunity for automation of ground truthing for classification tasks. For example, consider two documents, one being an ideal document and the other containing noise from binarization, crumpling, stains, or bleed through. Changes detected between these two documents that have identical content apart from the noise, could be used to train a detector for noisy regions in documents. This alignment also provides the ground truth necessary to learn the transformation from a noisy region to a clean region. Given enough correspondences from a large heterogeneous collection, it may be possible to provide a robust solution for noise removal in complex documents and outperform researcher made filters for some types of noise. Datasets could be created by focusing on one type of change (e.g. crumpling a sheet of paper), and once the technique is shown to be successful extended to more complex collections like the Tobacco dataset.

Chapter 4: Local Features for Writer Identification and Retrieval

Handwriting is a behavioral biometric, which captures the neuromuscular process of a person's ingrained stroke formations as viewed through the output of fine motor control muscles in the hand. Using handwriting as a biometric is challenging due to the large amount of variation that can occur in stroke formations as well as variances that occur from emotional and environmental factors. Our work focuses on offline writer identification, which uses handwriting samples which have already been captured as static document images as is common in heterogeneous document collections. This is distinct from a large body of work in online writer identification, which dynamically captures much richer information of the writer's movements through a pressure pad. Given a new handwriting sample, the goal of writer identification is to determine the author from a set of previously known writers. Writer retrieval on the other hand, assumes that there is one sample from a known writer and searches a large volume of handwriting samples with unknown authors and to create a ranked list based on similarity. The goal of our work is to introduce more powerful features to increase the performance of writer identification and retrieval systems in large heterogeneous document collections.

4.1: Related Work

Offline handwritten writer identification is a well-studied topic that has seen steady progress in the last ten years. Table 13 summarizes the performance of some of the previous literature on this topic. Please note that there is a large variance in the size and difficulty of the datasets used, so the accuracy is not directly comparable.

Author	Dataset Language	# of Writers	% Correct
Srihari [88]	English	1500	87
Schlapbach [89]	English	50	94
Schlapbach [90]	English	100	98
Bulacu [91]	English	650	89
Fiel [92]	English	350	90
Schomaker [93]	Dutch	250	87
Bulacu [94]	Arabic	350	88
Abdi [95]	Arabic	82	90
He [96]	Chinese	20	80

Table 13: Performance of past writer identification approaches.

Previous research by Srihari et. al. [88] established the individuality of handwriting by showing writer verification rates of 96%, and writer identification rates of 87%, for a dataset of 1500 writers. They identified macro features that operated at the paragraph, line and word levels, as well as micro features (gradient, concavity, and structure), at the character level. The micro features significantly outperformed the macro features. While the results from the first large study in automated writer identification are very impressive, the dataset contains identical passages from all writers and requires manual segmentation, which is not practical in an automated real world scenario.

In [93], Schomaker and Bulacu model character allographs by creating a codebook of connected component contours (CO3) and matching using a bag of features model. The CO3 feature, was simply a set of 100 consecutive X, Y coordinates sampled from the contour. In [91] Bulacu models the curvature of characters by introducing the edge hinge, which models the relative angle of two line segments on a character's contour. They combine this method with the CO3, slant features, and run lengths to achieve an identification accuracy of 89% on the bench mark IAM dataset.

In [89], Schlapbach uses a sliding window to extract features from lines of text and builds a Hidden Markov Model for each writer. The author uses the log likelihood output from the Viterbi algorithm to rank users and achieves a 97% recognition rate on 100 writers from the IAM dataset. This work is extended in [90] where Schlapbach uses a Gaussian Mixture Model and achieves an identification rate of 98.5% on 100 writers. Both of these techniques assume perfect line segmentation and require a substantial amount of training. The author uses a 4-fold cross validation on extracted lines during the experiments instead of entire pages, potentially mixing training and testing samples that occurred from the same page. Subsequent papers have used a leave-one-out methodology using between 300-650 writers in the experiments, as has been done in our experiments.

More recently, Fiel shows that SIFT features capture local shape and texture useful for writer identification [92]. He achieves a 90.8% Top-1 identification accuracy on the IAM dataset and extends the approach to retrieval. The authors of [96], [95], [94] extended writer identification to Chinese and Arabic. In [96] the authors use Gabor wavelets for features and HMMs to classify Chinese with 80% accuracy. In [95] and [94] the authors use shape features for Arabic datasets. [95] achieves a recognition rate of 90% on a dataset of 40 writers and [94] achieves an identification rate of 88% when using five training samples on a 350 writer dataset.

4.2: Local Features for Writer Identification

We believe that more powerful features that represent the individuality of handwriting can be extracted. Macro features such as slant and baseline are very useful for determining if two samples did not come from the same writer, but are not discriminative enough for finding an individual writer in a large collection. Local features that capture texture such as SIFT, LBP, or Run lengths can effectively discriminate writing style, but also lose important local information related to the character structure and stroke. Additionally they are potentially vulnerable to changes in the writing utensil such as the same writer using a pen versus a pencil. The edge hinge and CO3 features are the closest to ours, but we feel that both of these methods can be improved upon. The edge hinge method only takes into account the angle between two edge fragments and only does so within a very small local neighborhood of 5-10 pixels. This approach could potentially be generalized to multiple consecutive fragments of arbitrary length as we have done using the K-Adjacent Segments feature. The CO3 features were one of the first attempts in an automated allograph based feature, but the extraction of features from the contour did not handle the segmentation problem since connected components were used and the approach was sensitive to small variations present in handwriting since the X, Y coordinates from the contour were directly used as features. Hence, we present an approach to first attempt to segment characters and introduce a more discriminative contour gradient descriptor that captures local shape and curvature present in the allograph.

4.2.1: Writer Identification using K-Adjacent Segments

4.2.1.1: K-Adjacent Segments (KAS)

K-adjacent segments were introduced by Ferrari [97] as a feature to represent the relationship between sets of neighboring edges in an image for object detection. It has since been successfully extended for a number of applications in handwritten text including language identification [98] and text zone detection and classification [99] based on the feature's ability to capture discriminative local stroke information in document images. This work aims to generalize the edge-hinge feature used in [91] by modeling the character contours using a codebook of KAS features.

In order to extract KAS features from a document image, a set of edges must be found. In color or gray scale images, Ferrari uses a Canny edge detector. Document images are typically binary, so contours that capture the shape and curvature are extracted. A line fitting algorithm is then used to decompose the curves into a set of lines. This process is illustrated in Figure 39.

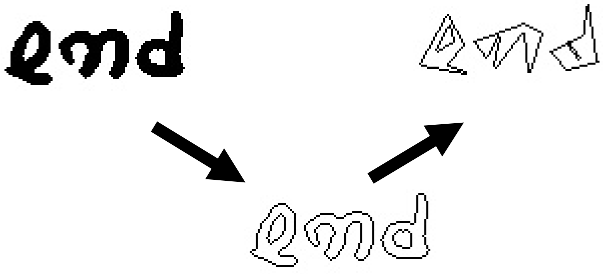


Figure 39: This image illustrates how contours and edges are extracted from connected components in documents.

As the name K-adjacent segments implies, this feature describes any number of K neighboring line segments, but for this work only 2, 3 and 4 adjacent segments (2AS, 3AS, 4AS) are tested. Any two lines are said to be adjacent if they share an endpoint. The lines that make up the KAS feature must be ordered in a consistent and repeatable manner so that KAS features can be directly compared against each other. The primary line segment is defined as the line with its midpoint closest to the center of the midpoints from all the lines. The remaining lines are ordered by their midpoints from left to right and then top to bottom. Each of the K lines can then be described by the following features:

$$\frac{r^{x_2}}{N_d}, \frac{r^{y_2}}{N_d}, \dots, \frac{r^{x_k}}{N_d}, \frac{r^{y_k}}{N_d}, \theta_1, \dots, \theta_k, \frac{l_1}{N_d}, \dots, \frac{l_k}{N_d} \tag{19}$$

Here (r^x , r^y) define the vector that connects the midpoint of a given segment and the midpoint of the primary segment. Θ and l are the orientation and length of a given segment that makes up the KAS feature. N is the length of the largest segment and is used as a normalization factor to make the feature scale invariant. Features for a 3AS are illustrated in Figure 40. Two KAS features, A and B, can be compared using the distance function D(A,B):

$$D(a,b) = w^r \sum_{i=2}^{k} \left\| r_i^a - r_i^b \right\| + w^\theta \sum_{i=1}^{k} \left| \theta_i^a - \theta_i^b \right| \ + w^l \sum_{i=1}^{k} \left| \log(l_i^a / l_i^b) \right| \qquad (20)$$

The weights w^r, w^θ, and w^l can be adjusted to assign more importance to particular features as needed. For this work we use weights of w^r=4, w^θ=2, and w^l=1 as done in the original paper [97] because the segment size is the least stable portion of this feature.

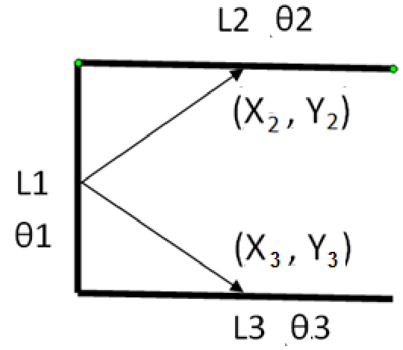


Figure 40: Segment ordering and features captured for a KAS, with the primary segment numbered 1.

4.2.1.2: Building a Codebook of KAS Features

A bag of features (BOF) model is used to compare the writers from two documents by converting the KAS features extracted from a document into a vector of code words. We use a clustering technique known as affinity propagation [100] to cluster KAS features from a set of training data to construct a codebook for the BOF model. The input to the affinity propagation algorithm is a distance matrix between all features. Initially all points are considered exemplar clusters and each cluster is combined with neighboring clusters using a message passing algorithm. Two types of messages are passed that represent the responsibility and availability for a given exemplar. The responsibility message, sent from point *i* to point *k*, is defined by *r(i,k)* and represents accumulated evidence for how well suited a point *i* is to be an exemplar for point *k*. The availability message, sent from point *k* to point *i*, is defined by *a(i,k)* and represents how appropriate it would be for point *i* to represent the exemplar of point *k*. The equations for both can be seen below.

$$r(i,k) \leftarrow s(i,k) - \mathrm{max}_{\mathrm{k}',\mathrm{k}'\neq \mathrm{k}} \{a(i,k') + s(i,k')\} \tag{21}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) - \sum_{i',i'\neq\{k,i\}} max\{0, r(i',k)\}\} \tag{22}$$

These messages continue to pass until a "preference" threshold is met. It should be noted that unlike K-means this algorithm does not require the number of clusters ahead of time and that the number of clusters is instead controlled by the preference threshold. Once a codebook is constructed, the source document is represented by a feature vector of KAS "code words" present in the document. This feature vector is normalized to sum up to 1 so that it is invariant to the size of the input. The two feature vectors can then be compared by their Euclidean distance. Figure 41 shows examples of the 20 most popular 3AS code words present in the IAM dataset.

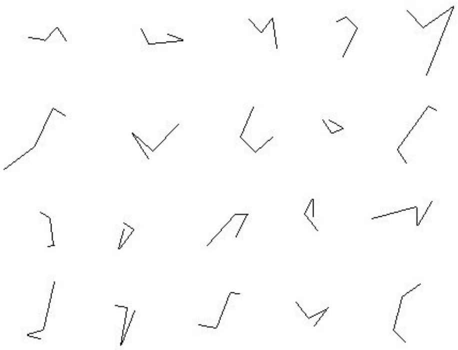


Figure 41: Example of TAS code words.

4.2.2: Writer ID with an Alphabet of Contour Gradient Descriptors

When comparing two handwriting samples, forensics document examiners typically match specific attributes from corresponding characters that are invariant to normal variation in handwriting. Examples of such attributes include the shape of loops, the curvature of letters, and start and end strokes. While the character matching approach used by forensic document examiners would likely be an improvement for algorithms attempting to automate writer identification, this general approach is hindered by the fact that handwritten character segmentation and recognition remain open research problems. Hence, the “illiterate" algorithms developed thus far largely rely on global features such as slant and run lengths, or aggregated histograms of local features from patches or sub-portions of the contour. In contrast, this approach attempts to emulate the approach taken by forensic document examiners and make the assumption that a segmentation, which extracts repeatable regions, can be substituted for the optimal character segmentation. A novel contour gradient descriptor designed for binarized character-like segments, which capture local shape and curvature unique to individual writers, is introduced. These features are first clustered into a pseudo-alphabet for each writing sample. A unique distance measure, which calculates the character similarity between two alphabets, is then used to determine writer similarity.

4.2.2.1: Extracting Character-Like Segments

The three segmentation strategies described below are used in our work. We know that these segmentation schemes are not considered state of the art in character segmentation, but the intent is to show that writer identification can be performed with relatively simple segmentation schemes as long as repeatable segments are extracted. Please note that the segmentation assumes a binarized document image.

Connected Components

The first segmentation scheme simply takes all connected components from a binarized writing sample. This segmentation strategy can be considered near optimal for a print script where few characters are touching or when writers have characters that touch consistently. However, in cases where there is a cursive script this approach will likely either capture words or several connected characters as components. While this is not optimal, previous research has shown that the shape of full words are discriminative enough to be used for writer identification [101].

Vertical Cuts

In an attempt to find more character-like and repeatable segments, a second segmentation scheme is used to attempt to splice large connected components into repeatable pieces. Each of the black pixels is assigned an energy, E(x,y), which is calculated by the following:

$$\mathrm{E(x,y)} = \begin{Bmatrix} A & if\ I(x,y) = 1 \\ log(y+B) & if\ I(x,y) = 0\ and\ \mathrm{C(x,y)} = 0 \\ C & if\ I(x,y) = 0\ and\ \mathrm{C(x,y)} = 1 \end{Bmatrix} \quad (23)$$

Here I(x,y) is the pixel intensity (1=white, 0=black). C(x,y) equals 1 if the pixel falls on the contour (edge) and is otherwise 0. The reason contours are assigned a larger energy than other black pixels is to discourage the segmentation of loops, which are known to be discriminative for writer identification. The energy function for non-contour black pixels uses the log of the pixel's vertical displacement in order to create lower scores for cuts near the bottom of a connected component, which is known to correspond with many character segments in cursive script. So long as the following relationship is maintained, A<B<<C, we do not see a significant difference in the quality of cuts made. An energy, E(x), is calculated for each column as follows in equation (24), where H is the height of the column in pixels:

$$E(x) = \sum_{i=0}^{H} E(x,i) \tag{24}$$

Next, a sliding window is used to select the column with the lowest energy in which to make the segmentation cut. Once a cut is selected the sliding window is placed at the pixel column following the cut. This is repeated until the entire connected component is traversed. In order to generalize the window to writing styles with varying sizes, the window width is set to a percentage of the median connected component height from a given sample. In order to prevent the sliding window from continuously selecting small components, the first ¼ of the sliding window is not used. The vertical cuts segmentation and tracking window are illustrated in Figure 42.

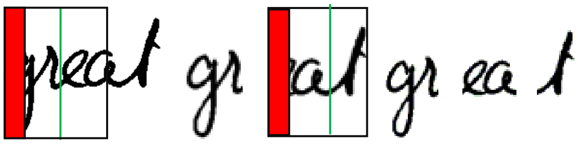


Figure 42: Two iterations of the sliding window. The red area shows unused parts of the window. The green line corresponds to the segmentation.

Seam Cuts

The final segmentation approach uses heuristic path planning similar to seam-carving, which chooses the path with the lowest energy on which to make the segmentation [102]. This approach has the advantage that it can accommodate segmentation cuts around curved strokes or slanted characters. For a connected component from a binary image we use the same energy function E(x,y) as defined in the previous section. For segmentation we define a possible seam to be a path from a given pixel at the top of the connected component to any pixel at the bottom. More formally we define the energy for a seam given starting coordinates x,y to be:

$$\mathrm{M}(\mathrm{x},\mathrm{y}) \ = \ \mathrm{E}(\mathrm{x},\mathrm{y}) + \min(\mathrm{M}(\mathrm{x}+1,\mathrm{y}+1), \mathrm{M}(\mathrm{x},\mathrm{y}+1), \mathrm{M}(\mathrm{x}-1,\mathrm{y}+1)) \qquad (25)$$

This energy calculation can be programmed very efficiently with dynamic programming. In order to find a seam path, one only needs to backtrack the path taken during the energy calculation. All seams are found from top to bottom of connected components and if seams overlap, then the seam with the lowest energy is retained. Finally, the sliding window from the previous section is used again to choose the seam with the lowest energy to make the segmentation. This approach is outlined in Figure 43.

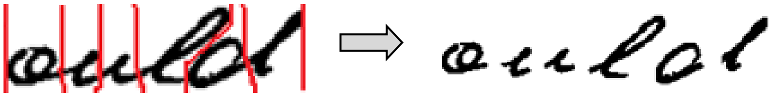


Figure 43: Seam cut example.

4.2.2.2: Contour Gradient Descriptor

Recently, features that extract local properties of character contours have been shown to be the most effective for writer identification [91], [94], [15]. We propose a novel descriptor for binary characters, which captures the shape and curvature of a character-like segment and is shown to significantly outperform previous descriptors. In order to calculate the descriptor, we first extract the contour from the binarized region. For each point on the contour, the gradient is calculated by taking two contour segments of size P from either side of the point and calculating the combined slope as shown in Figure 44. In order to make the implementation robust to scale changes, the size of the segment P is determined as a percentage of the median height of connected components from a given writing sample. Using longer segments or larger values for P to calculate the gradient of the contour can be viewed as a smoothing factor for noise from the binarization.

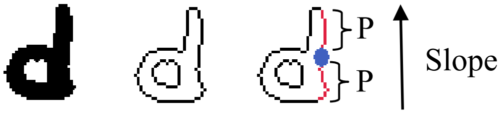


Figure 44: Handwritten letter contour and the slope calculation at each point (blue circle) per contour segment (red line).

If the contour is followed in a clockwise manner when calculating the gradient, then the full 360 degrees can be assigned to the gradient values. Next a SIFT-like [62] descriptor is created by placing an NxN grid on top of the contour gradients. The grid is stretched horizontally or vertically to fit non-square regions. A histogram of gradient orientations is calculated by binning the gradients into eight orientations (0º, 45º, 90º, etc.) for each region as shown in Figure 45. Finally, the descriptor is normalized by dividing each of the dimensions by the sum of the total gradient energy. We found the L1 distance measure and a value of 4 for N to give the best results for the descriptor. In order to compensate for variance in component widths in the distance measure, the score for two segments is set to be the maximum possible distance if the aspect ratios for the segments are not within one log scale of each other.

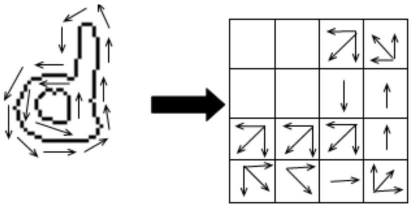


Figure 45: Contour gradients and the resultant feature.

4.2.2.3: Assigning a pseudo-alphabet

Given a set of local features, a typical approach in writer identification thus far has been to create a global descriptor by fitting them into a bag of features (BOF) framework [91], [92], [15]. However, a significant amount of information can be lost when assigning these high dimensional local features to a relatively low number of cluster centers or codewords, especially if the data that the code words are built from do not adequately represent the feature space. This effect is especially undesirable in writer identification, where we want to capture small variations in writing style that are lost when features of extracted patches or segments are assigned to codewords. Furthermore, codebook approaches capture information about the language such as the character frequency that is not desirable for writer identification since it will over represent common characters. Professional document examiners certainly do not compare writing samples by measuring the distance of letters to a reference guide, but rather directly compare matching letters from two writing samples with each other.

Thus, instead of creating a global codebook, the extracted character-like segments are clustered to form a pseudo-alphabet for each writing sample and the feature closest to the cluster centers are retained as exemplar "letters" for the writer. While one could use each "letter" or extracted segment from the writing sample, the clustering is done to prevent commonly occurring segments, such as the letter “e”, from dominating the distance measure. K-means of the contour gradient descriptor is used to perform the clustering. Since the true number of clusters is unknown due to variation in the segmentation, a sufficient value for k is found experimentally and little improvement is seen beyond 150 clusters. In cases where k is greater than the number of segments, all segments are retained. An example of an extracted alphabet from one of the writing samples can be seen in Figure 46.

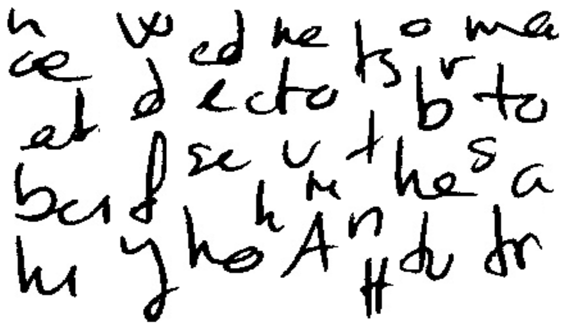


Figure 46: Example of pseudo-alphabet from extracted segments.

Given two writing samples and their associated pseudo-alphabets, an asymmetric distance measure to allow matching alphabets of different sizes can be calculated as follows:

$$D(A,B) \; = \sum_{i=0}^{N_A} min(\,|A_i - B_0|, \quad \ldots \; , |A_i - B_{N_B}|) \; / N_A \qquad (26)$$

Here A, B are alphabets created from two handwritten samples, A_i, B_i are the contour gradient features for “letters” in the alphabet, and N_A, N_B are the number of clusters in each alphabet. This sum is the total distance measure, which calculates the distance between the closest pair of letters in each alphabet. In other words, it is the minimum distance required to transform alphabet A into alphabet B.

A likely application for writer identification in law enforcement would be to search for a sample of a known author against a database of handwriting samples from unknown authors or vice versa. For this application, it would make sense to take advantage of statistical distributions of the handwriting samples across the database. For example, certain characters, such as the letter “o”, are constructed the same by many authors. In particular, it is natural to borrow the concept of inverse document frequency (IDF) from information retrieval to increase the significance of query letters that occur less frequently. Since the underlying features are not assigned to codewords in common between alphabets, the IDF is found dynamically at query time by summing the number of alphabets that have at least one cluster center within a threshold t of the query letter and taking the log of that value. Hence, the distance measure can be updated as:

$$D(A,B) \; = \sum_{i=0}^{N_A} w_i * min(\,|A_i - B_0|, \quad \ldots \,, |A_i - B_{N_B}|)/N_A \qquad (27)$$

Where,

$$w_i = \; log(\frac{\#\, of\ Samples(S)}{\sum_{j=0}^{S} count\ (abs(A_i - S_j) < t)}\ ) \qquad (28)$$

4.2.3: Experiments

Evaluations were conducted on the IAM and ICDAR 2013 datasets to determine the effectiveness of KAS and Contour Gradient Descriptor features for writer identification.

4.2.3.1: IAM Dataset

The IAM dataset [103] consists of handwritten English text from 651 different writers. Each sample is made up of two or three sentences. 159 writers provided three or more samples, 142 writers provided two samples and the remaining 350 writers provided a single sample. This dataset has been used by a number of other authors [89], [104], [15], [92] and can be considered the benchmark dataset for writer identification. In order to process the gray scale images, each image is preprocessed by binarizing the data using a threshold of 70%. Figure 47 illustrates samples from two different writers.

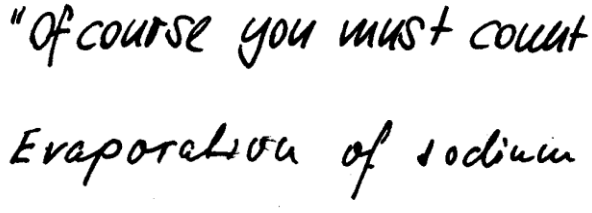


Figure 47: Two writer samples from the IAM dataset

Two random samples from 301 writers are used. Recognition is performed using K-nearest-neighbors (KNN) in a leave one out manner, meaning each image was compared against the 601 other documents with only one possible positive match. The results for this experiment are shown in Table 14, along with previous state of the art approaches. These results indicate that 3AS is the best feature representation for K-Adjacent Segments, with a Top 1 recognition rate of 93.3%. This could be because 2AS does not capture as much information and there were less repeatable 4AS features found in a given document. While the features are similar to the edge hinge and slant features used previously in [91], the improved performance is likely due to the extra segment found in the 3AS feature, the addition of segment size in the feature representation, and the use of a codebook of clusters rather than coarse quantization. Given the superior performance of the 3AS features, it is used for the remaining experiments.

Using vertical cuts provides for a nearly 50% reduction in the Top-1 error rate over the KAS features and significantly outperforms all previous approaches. A close examination of the errors revealed that five of the writers changed their writing style, including one of them signing a different name. This made it very difficult to correctly identify these writers and resulted in an error rate of 1.7%. Perhaps the most surprising result is that segmentation using only connected components performed at a Top-1 rate of 91.8%, which is already comparable to the state of the art on this dataset.

IAM Dataset Results	Top-1	Top-5	Top-10
Connected Comp.	91.8	93.8	94.0
Seam Cuts	95.4	96.7	97.3
Vertical Cuts	**96.5**	**97.2**	97.3
KAS – 2 AS	89.6%	94.0%	94.6%
KAS – 3 AS	**93.3%**	**95.3%**	96%
KAS – 4 AS	92.0	95.0	95.8
SIFT [92]	90.8%	96.5%	**97.5%**
Edge-Hinge + CO3 [91]	89%	N/R	96%

Table 14: Writer ID Accuracy (%) on the IAM dataset

Further inspection of the handwriting styles in the dataset showed that approximately 31% of the data is cursive script, 38% is a mix of both cursive and print scripts, and the remaining 31% of the data is print script. This means that the contour gradient feature can discriminate between writers even using partial and full words, and not just single characters. Both the vertical and seam cuts improved the Top-1 identification rate by 3-4% reducing the error rate by nearly 50%. While it was expected that seam cuts would outperform the vertical cuts, this discrepancy could be explained by the lack of large slant in most handwriting styles in the dataset.

4.2.3.2: 2013 ICDAR Writer ID Contest

We participated in a recent contest for writer identification held at the 2013 ICDAR Conference [17]. Each participant was required to provide an executable program to the organizers that provided a distance between two handwriting samples, which the organizers used to evaluate each participant's submission on a dataset consisting of 4 samples (2 Greek and 2 English) from 250 previously unseen writers. Following the contest, this dataset was made public for future research. 12 systems participated, including implementations of approaches that were previously state of the art. Our approach discussed in Section 4.2.2, using an alphabet of contour gradient descriptors, placed first at the contest. We summarize the results from two of the experiments conducted by Louloudis et. al. during this contest and the full evaluation can be read in [17].

Greek Results	Top-1	Top-2	Top-10
Seam Cuts	95.6%	98.2%	99.2%
Vertical Cuts	95.2%	97.6%	99.0%
KAS-3AS	86.0%	90.6%	96.4%
SIFT + SOH [105]	93.8%	96.4%	97.8%
SIFT + Fisher Vector [106]	88.4%	92.0%	97.8%
Run-length +Edge Hinge [107]	92.6	96.0%	98.4%

Table 15: *Accuracy (%) on 250 Greek writers from the ICDAR 2013 writer ID contest. Table adapted from [17].*

English Results	Top-1	Top-2	Top-10
Seam Cuts	94.6%	97.0%	98.8%
Vertical Cuts	94.4%	96.6%	99.0%
KAS-3AS	86.4%	90.4%	96.0%
SIFT + SOH [105]	92.2%	94.6%	96.8%
SIFT + Fisher Vector [106]	91.4%	94.2%	97.2%
Run-length +Edge Hinge [107]	91.2%	93.4%	96.6%

Table 16: *Accuracy (%) on 250 English writers from the ICDAR 2013 writer ID contest. Table adapted from [17].*

Their first experiment uses 250 Greek handwriting samples and tests them in a leave one out manner as is done with the IAM dataset. The second experiment follows the same procedure with the 250 English handwriting samples. The top 1, 2, 10 identification metrics are reported, which measures if the correct match is found in the top-N results. Table 15 and Table 16 show the results and compares our approach to the top algorithms at the contest. These results further validate the positive results on the IAM dataset, showing a reduction of error over other approaches of about 30%. The seam cuts slightly outperformed vertical cuts on this dataset.

4.3: Combining Local Features for Writer Identification

In recent years, several new and powerful local features proposed for writer identification have significantly boosted performance over previous methods. Our hypothesis is that combinations of these features will outperform the individual features since they capture different attributes of handwriting and therefore, should be complementary to one another. Here, we focus on three types of features: features produced from segmentation-free methods such as SIFT or SURF, that extract features from interest points; edge-base features extracted from character contours; and features from allograph methods that aim to capture a character's shape and style. The Fisher Vector is used for feature pooling and a linear combination of distances is then used to combine the features.

4.3.1: Local Features

We use three local features (KAS, SURF, and CGD) that have demonstrated strong writer identification performance for feature combination. These features were chosen because they capture different attributes of handwriting and should boost performance when combined. The KAS and CGD features are reused from 4.2.1 and 4.2.2, though we embed the KAS feature in to a L2 normalized feature vector as shown in Equation (19). SURF was chosen because a number of recent papers ([108], [105], [92]) have shown the effectiveness of interest point based methods for writer identification. The advantage of these methods is that they do not require any binarization or segmentation of the document image, while still capturing local texture and shape. We tried several interest point based methods and found that the OpenCV implementation of SURF [61] outperforms SIFT [62], especially in datasets containing only binarized images.

For each document image, a set of interest points is extracted using the Fast Hessian detector as shown Figure 48. For each interest point, a 64-dimensional SURF feature vector is extracted by splitting the patch into a 4x4 grid and extracting (∑Dx, ∑Dy, ∑ |Dx|, ∑|Dy|), where Dx and Dy are the Haar wavelet response in the x and y directions.

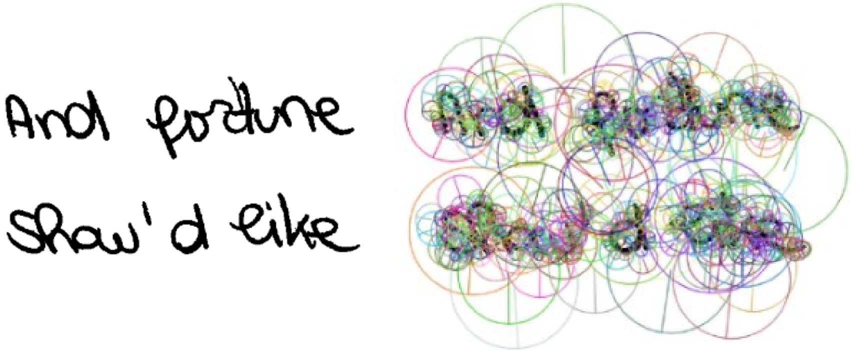


Figure 48: SURF Features Extracted from Handwriting

4.3.2: Feature Pooling

Vector quantization is often used to create codebooks that aggregate local features into a bag of words representation. However, this method suffers from coarse quantization and only captures histogram counts, losing higher order statistics. Fisher Vectors [14], introduced by Perronnin, have become popular for object recognition and retrieval because they address some of these concerns. We were strongly motivated to use Fisher Vectors to aggregate local features for two reasons. First, the work by Fiel et. al. in [106] showed significant improvement in writer identification over a bag of words when using the Fisher Vector for SIFT features. Second, Fisher Vectors also share a close derivation with Gaussian Super Vectors [109], which have shown state of the art results for speaker identification.

In order to generate a Fisher Vector, a Gaussian Mixture model ($u_\lambda(x)$) must first be created from training data as shown in Equation (29). This can be viewed as a generative model for the local features and shares the motivations for universal background models previously used in speaker and writer identification.

$$u_\lambda(x) = \sum_{k=1}^{K} w_k u_k(x) \tag{29}$$

$$u_k(x) = \frac{1}{2\pi^{D/2} * |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-u_k)' * \Sigma_k^{-1} * (x-u_k)} \tag{30}$$

A simple probabilistic bag of words model then accumulates histogram counts using the soft assignment of feature $x_t$to the Gaussian *k,* as shown in Equation (31).

$$\gamma_\lambda(k) = \frac{w_k u_k(x_t)}{\sum_{j=1}^{K} w_j u_j(x_t)} \tag{31}$$

$$G_{u,k}^{X} = \frac{1}{T\sqrt{w_k}} \gamma_\lambda(k) \left(\frac{x_t - u_k}{\sigma_k}\right) \tag{32}$$

$$G_{\sigma,k}^{X} = \frac{1}{T\sqrt{2 * w_k}} \gamma_\lambda(k) \left(\frac{(x_t - u_k)^2}{\sigma_k^2} - 1\right) \tag{33}$$

Fisher Vectors, on the other hand, accumulate the partial derivatives with respect to the mean and variance parameters as shown in Equations (32) and (33). The intuition is that by accumulating the gradient with respect to the Gaussian parameters, one is capturing how much the background model has to change to account for the newer local features. The Fisher Vector is the concatenation of the K gradient vectors from $G_{u,k}^{X}$ and $G_{\sigma,k}^{X}$ leading to a large 2*K*D feature vector (where D is the dimension of the local feature).

In practice, two improvements to Fisher Vectors are used to increase performance. The first is power normalization, which involves taking the square root of each dimension to discount frequent features. The second is L2 normalization of the Fisher Vector to account for cases in which there are a different number of local features per sample. The cosine distance is also shown to be the natural method to compare two Fisher Vectors [14]. The use of Fisher Vectors improved the writer identification performance for KAS and CGD features in comparison to the codebooks and cluster comparison approaches used in in the previous sections so it was used for pooling all three local features.

4.3.3: Feature Combination

Given the normalized Fisher Vectors for each of the local feature types, a linear combination is used to fuse the distances from each feature as shown in Equation (34), where A and B are two samples being compared, FV is the Fisher Vector for feature type k (e.g. KAS, SURF, or CGD), and $w_k$is the weight for each sample.

$$D(A,B,K) = \sum_{k=1}^{K} w_k * \left(FV(A_k) \bullet FV(B_k) \right) \qquad (34)$$

Where,

$$1 = \sum_{k=1}^{K} w_k \qquad (35)$$

Given a training dataset in the experiments, we performed a grid search to determine the optimal values for w_k for each feature type, by selecting the parameters for $w_k$that have the largest MAP when evaluating the training set in a leave-one-out manner. These learned weights were then used during the evaluation of the test set. For comparison, we also reported a simple linear combination using equal weights for all three features in our experiments; i.e. w_k is always set to 1/3.

4.3.4: Experiments

The effectiveness of the feature combination approach was evaluated on four datasets: IAM, ICDAR 2013, CVL, and MADCAT datasets. For each of the experiments the number of Gaussians used in the mixture model to create the Fisher Vectors was set to 64 since little improvement is seen beyond this point. The GMM and feature combination weights are always trained using an alternate dataset to avoid mixing writers in the training and testing data. For example, the IAM experiment used the CVL dataset for training and vice versa. The ICDAR 2013 experiment was trained on samples from the ICFHR 2012 contest, which was available to contestants and also contained Greek handwriting. The MADCAT experiment used the training and testing split described below.

In the results, the methods are abbreviated as follows: K-Adjacent Segments (K), SURF (S) and Contour Gradients (C). The combination of the three features using equal weights is denoted by K&S&C, whereas the weights found from training are denoted by K&S&C*. The results generated from our approach are highlighted in blue, while comparable results from other papers are left with a white background. The top performing methods are highlighted in bold.

4.3.4.1: Evaluation Metrics

The datasets were tested in a leave-one-out manner meaning one image is taken out from the test dataset and queried against the remaining documents. For each of the experiments, the soft criterion for the Top-N results was used. This measures if at least one document from the same writer was found in the first N results. For the CVL and MADCAT datasets the hard criterion, which indicates if the correct writer is found in all of the top N ranked results, is also reported since there are more than two samples for each writer. These metrics and procedures are consistent with previous studies ([104], [17] , [105]). To provide a more complete picture of the retrieval performance, Mean Average Precision (MAP) is also provided.

4.3.4.2: IAM Dataset

We again use the IAM dataset similar to Section 4.2.3.1. 301 writers provided at least two samples and the remaining 356 writers provided only a single sample. However, in order to have our results comparable to more recent publications, we took the first two samples from each of the 301 writers, and split the single samples from the other 356 writers in order to create a dataset with 657 writers containing two samples each. This slightly dropped the performance of the approaches due to the limited amount of handwriting available in the split cases.

Features	Top-1	Top-2	Top-5	Top-10	MAP
K	88.8	91.1	95.0	96.4	0.914
S	90.0	92.4	96.2	97.6	0.926
C	91.3	93.8	96.6	97.6	0.936
K&S&C	94.1	96.0	98.2	98.5	0.958
K&S&C*	94.7	95.9	98.1	98.7	0.960
Chain Code [110]	91	N/R	N/R	97%	N/R
Edge+ CO3 [104]	89	N/R	N/R	96%	N/R
SIFT+SOH [105]	**98.5**	**N/R**	**99.1**	**99.5**	**N/R**

Table 17: Results on the IAM Dataset

The experimental results for the IAM dataset can be found in Table 17 along with a comparison to other existing methods. The individual features each perform well, but when combined they further reduce the MAP error rate by 37% over the top performing feature. The optimized weights perform comparably to the naïve equal weights, largely due to the fact that all three features contribute to the boost in performance. While the feature combination outperforms most existing systems, it is unable to outperform the nearly perfect results on this dataset reported by [105] using a SIFT based approach.

4.3.4.3: ICDAR 2013 Writer ID Contest

We again use the ICDAR 2013 Writer ID competition dataset similar to Section 4.2.3.2. The results for the Greek and English experiments are shown in Table 18 and Table 19. Again the individual features under evaluation all perform very well on the ICDAR 2013 dataset. The Fisher Vector significantly boosts performance of the KAS feature over the codebook based approach submitted to the competition. For this dataset, the KAS feature only slightly improves performance when using the equal weighted feature combination. For this reason the trained weights, which discounts the KAS feature, reduces the error rate of the individual features by over 60%. On the English dataset there is a smaller 30% reduction in the error rate over the individual features, but both combined approaches again perform well beyond existing methods including the SIFT+SOH, which had the best performance on the IAM dataset.

Features	Top-1	Top-2	Top-5	Top-10	MAP
K	93.2	95.6	98.0	99.0	0.952
S	94.6	97.2	98.8	99.2	0.964
C	97.2	98.6	99.2	99.6	0.984
K&S&C	98.2	99.0	99.4	99.8	0.988
K&S&C*	**99.2**	**99.6**	**99.8**	**99.8**	**0.995**
SIFT+FV [106]	88.4	92.0	96.8	97.8	N/R
SIFT+SOH [105]	93.8	96.4	97.2	97.8	N/R
Edge + Runs [107]	92.6	96.0	98.0	98.4	N/R

Table 18: Results on the ICDAR 2013 Greek Dataset

Features	Top-1	Top-2	Top-5	Top-10	MAP
K	92.4	94.4	96.4	97.2	0.942
S	94.6	96.2	97.6	98.0	0.959
C	96.4	97.2	98.0	98.6	0.971
K&S&C	97.0	97.8	98.0	98.6	0.976
K&S&C*	**97.4**	**97.8**	**98.6**	**98.8**	**0.979**
SIFT+SOH [106]	91.4	94.2	95.8	97.2	N/R
SIFT+FV [105]	92.2	94.6	96.4	96.6	N/R
Edge + Runs [107]	91.2	93.4	96.2	96.6	N/R

Table 19: Results on the ICDAR 2013 English Dataset

4.3.4.4: CVL Dataset

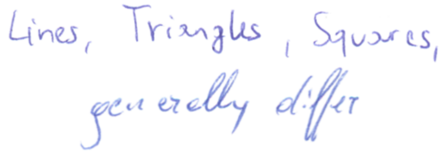


Figure 49: Two writer samples from the CVL dataset

The CVL dataset [111] was recently released to promote research in writer identification and word spotting. It consists of five passages for 309 writers, with four of the passages written in English and the fifth written in German. Figure 49 shows samples from two different writers.

Features	Top-1	Top-2	Top-5	Top-10	MAP
K	98.5	99.1	99.2	99.5	0.927
S	98.7	99.2	99.4	99.5	0.941
C	97.0	98.1	99.0	99.4	0.881
K&S&C	**99.4**	**99.5**	99.5	**99.7**	0.966
K&S&C*	**99.4**	**99.5**	**99.6**	**99.7**	**0.969**
SIFT+FV [106]	97.8	98.6	99.1	99.6	N/R
Edge + Runs [107]	97.6	97.9	98.3	98.5	N/R
Grid [112]	97.7	98.3	99.0	99.1	N/R

Table 20: MAP and Soft Criterion Results on the CVL Dataset

Features	Top-2	Top-3	Top-4
K	94.3	85.9	66.2
S	96.1	88.5	70.7
C	91.0	77.8	52.3
K&S&C	**98.3**	**95.2**	80.8
K&S&C*	**98.3**	94.8	**82.9**
SIFT+FV [106]	95.6	89.4	75.8
Edge + Run Length [107]	94.3	88.2	73.0
Grid [112]	95.3	94.5	73.0

Table 21: Hard Criterion Results on the CVL Dataset

The results for the CVL experiment can be found in Table 20 and Table 21 below. The main advantage of the feature combination can be seen in the MAP and hard Top-4 evaluations, where the error rate of individual features is reduced by over 30%. The best combination appears to be using all three features, with the majority of the performance gain coming from the KAS and SURF features. This could be due to the weakness of the Contour Gradient Feature if there is a substantial change in the allograph, which sometimes occurred between the English and German samples. Again the KAS feature is significantly improved by the Fisher Vector over the results reported in [111]. The combination of these three features also substantially improves over results reported by existing approaches for the soft and hard criterion.

4.3.4.5: MADCAT

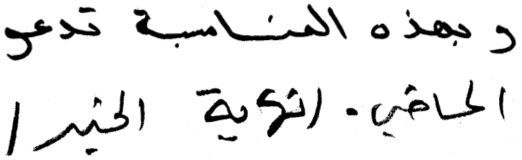


Figure 50: Two writer samples from the MADCAT dataset.

The DARPA MADCAT dataset [113] consists of over 10,000 binarized pages of handwritten Arabic text from over 325 writers. The images are sampled at 600 dpi, are already binarized, and are significantly noisier and less structured then the IAM dataset. For example, writers that contributed to this dataset were directed to write at various speeds using various writing instruments (pencils, pens and markers) and to add natural variation into the handwriting samples. We formed a dataset consisting of ten samples randomly drawn from 316 writers. This was split into a training set consisting of ten samples from sixteen writers to build the GMM and a test set using the remaining samples from 300 writers. Examples of the handwriting are shown in Figure 50.

Features	Top-1	Top-2	Top-5	Top-10	MAP
K	96.8	98.1	99.1	99.4	86.4
S	92.8	94.2	95.5	96.0	72.2
C	96.9	98.2	99.4	99.6	86.8
K&S&C	97.1	98.0	99.0	99.3	87.9
K&S&C*	**97.8**	**98.6**	**99.4**	**99.5**	**90.1**

Table 22: MAP and soft criterion Results on the MADCAT Dataset

Features	Top-2	Top-3	Top-5	Top-7	Top-9
K	93.3	90.5	82.2	68.4	39.7
S	87.1	82.1	69.5	51.8	17.0
C	93.2	90.1	80.9	67.6	40.1
K&S&C	94.2	91.4	86.6	76.3	43.3
K&S&C*	**95.4**	**93.2**	**87.5**	**78.0**	**50.9**

Table 23: Hard Criterion Results on the MADCAT Dataset

The MAP, soft and hard criterions for the MADCAT experiment can be found in Table 22 and Table 23. In this case, the KAS and CGD features significantly outperform SURF, which can be seen clearly in the MAP and Hard Top-N evaluations. This could possibly be from less discriminative patches being extracted due to the elongated nature of Arabic script, and the bias of the SURF keypoint detector to extract circular regions. Due to the relatively poor performance of SURF, the naïve feature combination only slightly outperforms the KAS and CGD features. The trained feature weighting nearly discounted the SURF features, decreasing the error rate by 30% and also increasing the hard Top-9 performance by 10%. Only one paper [114] has reported results for writer identification on the MADCAT data and they report an accuracy of 75% on similar data with 60 writers.

4.4: Conclusion

4.4.1: Summary

Our work has made several contributions in the area of writer identification by using more powerful local features for writer identification. The KAS feature improves upon past edge based approaches in which many features were combined and still did not reach the same level of performance. We have also shown an approach using basic segmentation, which mimics forensic handwriting examiners, and improves writer identification performance by extracting repeatable character-like segments. A feature based on contour gradients as well as a unique pseudo-alphabet based framework for matching these features was introduced. Furthermore, a weighted combination of both of these local features and SURF aggregated using

Fisher Vectors produces state of the art results. These methods improve upon the previous state of the art across 3 different scripts, reducing the error rate by as much as 50% on benchmark datasets. The results of our experiments also demonstrate that larger and more difficult writer identification datasets are needed, as these results are approaching perfection on current datasets.

4.4.2: Future Work

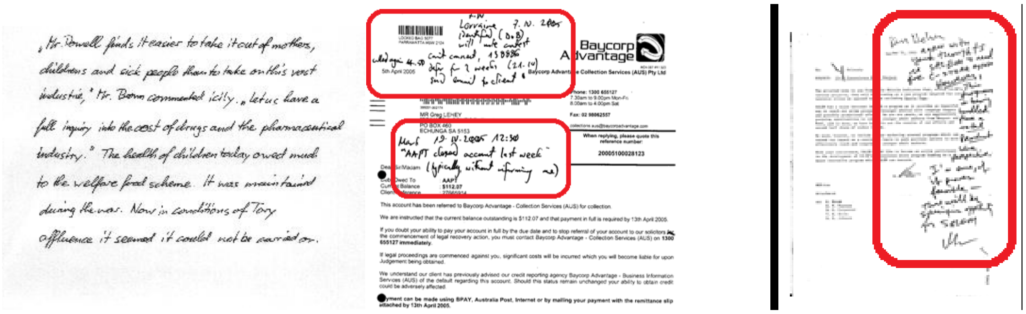
4.4.2.1: User Driven Writer Identification

Document image and pattern recognition researchers have generally taken approaches at two extremes for user involvement in writer identification. At one end of the spectrum, early work by Srihari [88] assumed that a user would individually segment and label each character from both the questioned document as well as any set of documents they wanted to compare against. This approach produced impressive results by having users manually solve the challenging segmentation and character recognition problems, but it would be very cumbersome and time consuming for an examiner to annotate many thousands of samples in this manner. On the other hand, more recent approaches have tried to completely automate the process given a handwritten sample of interest. The inability of current handwriting recognition algorithms to segment and identify characters with high precision has led to researchers using global features such as slant or aggregated local features from text lines [89], interest points [92], or connects components [15]. While these approaches are fully automated, even approaches that are state of the art, such as our pseudo-character approach presented in Chapter 4, likely give up some performance from the inability to take advantage of character labels.

One potential compromise would be to have a user only segment characters from the questioned document, without also having to manually segment characters from the potentially larger set that they had to compare against. In this scenario, an algorithm could be created that builds a writer model trained on each labeled character from a given writer and then each model is compared against a test set to see if this improves performance. This could be accomplished in a number of ways, such as having segmentation on new samples driven by a user's segmentation on a questioned document or a template match procedure based on sliding windows similar to current HOG [115] based approaches for object detection. In fact, Forensic Handwriting Examiners, who are now using commercial tools developed by document image researchers over the past few years, have also requested similar approaches. They find fully user driven segmentation approaches to be too time consuming on a large dataset, while fully automated approaches often return results that are hard to interpret.

4.4.2.2: Applications to Noisy Documents

The main datasets used for research in offline writer identification and retrieval usually come from ideal data sources consisting of only a plain white background and handwriting as shown in Figure 51. While testing on these idealized datasets is useful for research in designing features and classifiers, they do not always represent the variation present in real world collections. In such collections, images can often contain noise from aging, crumbling, stains, a mix of machine print, graphics, ruled lines or handwriting, and in some cases handwriting from several users. Several researchers have applied pattern recognition techniques to harder cases involving historical documents, but to the best of our knowledge, no one has looked at documents containing mixed content or multiple author's handwriting. In these cases, interesting problems involve comparing the performance of writer identification techniques on these more complex documents, especially when very little handwritten text is available.


„Mr. Powell finds it easier to take it out of mothers,
childrens and sick people than to take on this vast
industrie," Mr. Bonn commented icily. „let us have a
full inquiry into the cost of drugs and the pharmaceutical
industry." The health of children today owed much
to the welfare food scheme. It was maintained
during the war. Now in conditions of Tory
affluence it seemed it could not be carried on.
Baycorp Advantage

a) *b)*

Figure 51: Left: Handwriting sample from existing Writer ID datasets. Right: a more challenging example containing a mix of figures, machine print and handwriting.

One potential application would be to the documents residing in the National Archives, Supreme Court rulings, or records from presidential libraries. Handwritten notes on typed transcripts from a particular Supreme Court justice or presidential official are of interest to historians, since these documents provide a unique insight into a person's state of mind that is not always apparent in speech transcripts or official rulings. For example, could one find when the President, the Vice President, the Chief of Staff, or other advisors dissented from a certain policy? Can progression of neurological or physical handicaps such as Alzheimer's or Parkinson's, which manifests itself in handwriting, be followed? Given a large corpus, a historian would want to identify documents that contained handwriting and find all passages for a particular individual of interest even when their handwritten annotation may only consist of a few words or symbols. A challenge in pursuing this research would be in obtaining an appropriate dataset, since large collections that are not generated using the consent of authors are often hard to obtain due to privacy concerns.

4.4.2.3: Feature Learning

Recently hand crafted local features used in many domains including object recognition have been outperformed by deep neural network techniques that learn useful mid and high level feature representations [116]. This success has also been extended into similar biometric applications such as face recognition [117]. Thus far no one has applied deep learning methods to writer identification, but it would be interesting to see if this approach would be able to extract discriminative features between writers that also outperform existing approaches. One challenge would be to identify the regions from which to extract these features and we believe the pseudo-characters extracted in sections 4.2.2 would be a good candidate for this effort.

Chapter 5: Summary of Contributions and Publications

This dissertation presents novel research in the areas of document image retrieval, document image change detection, and writer identification that will enable systems to more effectively search through large heterogeneous document image collections. In this chapter we summarize the contributions and publications for the work presented in Chapters 2, 3, and 4.

5.1: Document Image Retrieval

5.1.1: Contributions

- We demonstrated that a segmentation-free image retrieval algorithm operates efficiently and performs well for document image retrieval tasks.
    - First application of SURF to binary document images. It is 4x more computationally efficient than SIFT in this setting.
    - Designed a novel hashing scheme that can efficiently index neighboring SURF feature vectors to scale our approach to 7 Million Documents and 40 Billion Descriptors.
    - Geometric verification used to accurately retrieve images such as logos using only one query image in contrast to previous methods using traditional machine learning with multiple training examples for each logo.
- We were the first to produce a study to directly evaluate image retrieval for user relevance on a large real world dataset of document images using OCR as a baseline.

- Image retrieval is shown to positively impact user relevance when used for relevance feedback.
- Demonstrated image retrieval outperforms text retrieval of OCR for a limited number of topics.
- Image retrieval is useful in cases when OCR quality is poor or little text is present.

5.1.2: Publications

1. Jain, Rajiv, and David Doermann. "Logo Retrieval in Document Images." Document Analysis Systems (DAS), pp. 135-139, 2012.
2. Jain, Rajiv, Douglas W. Oard, and David Doermann. "Scalable Ranked Retrieval Using Document Images" Document Recognition and Retrieval, SPIE Electronic Imaging, pp. 1-15, 2014. **(Best Student Paper Award)**

5.2: Change Detection for Document Images

5.2.1: Contributions

- We developed an accurate document verification technique that is invariant to common image transformations such as binarization, scale, and rotation as well as more challenging deformations that occur from camera capture of document images including perspective change, motion blur, and small curvature in the surface of the page.
- We were the first to look at the problem of change detection of document images. We introduced two techniques to detect local changes present at the word or character level that outperform OCR based techniques.

  - The first uses the Longest Common Subsequence algorithm on SIFT features extracted from the center of text lines.
  - The second introduces a segmentation free SIFT alignment to handle cases where line or page segmentation fail.

- We developed two datasets to enable future research into document image change detection.

5.2.2: Publications

1. Jain, Rajiv, and David Doermann. "VisualDiff: Document Image Verification and Change Detection." International Conference on Document Analysis and Recognition (ICDAR), pp. 40-44, 2013.
2. Jain, Rajiv, and David Doermann. "Localized Document Image Change Detection." (Submitted to ICDAR)

5.3: Writer Identification and Retrieval

5.3.1: Contributions

- We applied the K-Adjacent Segments feature to writer identification. At the time it was published this was the strongest feature for writer identification. It remains the strongest contour edge-based feature.
- We created an automated framework for writer identification that emulates forensic handwriting examiners, who directly compare allographs.
  - This method placed first in the ICDAR 2013 Writer ID Contest.
- We achieved state-of-the-art results from combining local edge, allograph, and keypoint features across several scripts including Greek, Latin, and Arabic.

5.3.2: Publications

1. Jain, Rajiv, and David Doermann. "Combining Local Features For Offline Writer Identification. International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 583-588, 2014.

2. Jain, Rajiv, and David Doermann. "Writer Identification Using an Alphabet of Contour Gradient Descriptors." International Conference on Document Analysis and Recognition (ICDAR), pp. 550-554. 2013.

3. Jain, Rajiv, and David Doermann. "Offline writer identification using K-adjacent segments." International Conference on Document Analysis and Recognition (ICDAR), pp. 769 – 773, 2011.

5.4: Other Publications and Patents

1. Shivashankar, Vikas, Rajiv Jain, Ugur Kuter, and Dana S. Nau. "Real-Time Planning for Covering an Initially-Unknown Spatial Environment." FLAIRS Conference, pp. 64-68, 2011.
2. Jain, Rajiv, and David C. Smith. "Method of neighbor embedding for OCR enhancement." U.S. Patent No. 8,938,118. 20 Jan. 2015.

Bibliography

[1] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman and J. Heard, "Building a test collection for complex document information processing," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 665-666, 2006.

[2] B. Klimt and Y. Yang, "The Enron corpus: A new dataset for email classification research," in *Machine learning: ECML 2004*, Springer, 2004, pp. 217-226.

[3] "Sarah Palin E-mails," [Online]. Available: http://documents.latimes.com/sarah-palin-emails/.

[4] "Digitization Project," 2013. [Online]. Available: http://www.archives.gov/digitization/strategy.html.

[5] "Google Books," [Online]. Available: http://books.google.com/.

[6] "National Archives UK," [Online]. Available: http://www.nationalarchives.gov.uk/.

[7] "European Union National Archives," [Online]. Available: http://ec.europa.eu/historical_archives/index_en.htm.

[8] R. Jain and D. Doermann, "Logo retrieval in document images," in *International Workshop on Document Analysis Systems*, pp. 135-139, 2012.

[9] R. Jain, D. Oard and D. Doermann, "Scalable Ranked Retrieval Using Document Images," *IS&T/SPIE Electronic Imaging. Document Recognition and Retrieval,* pp. 1-15, 2014.

[10] R. Jain and D. Doermann, "VisualDiff: Document Image Verification and Change Detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 40-44, 2013.

[11] R. Jain and D. Doermann, "Localized Document Image Change Detection," in *International Conference on Document Analysis and Recognition (Submitted)*, 2015.

[12] P. Dreuw, D. Rybach, C. Gollan and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 21-25, 2009.

[13] H. Cao, R. Prasad and P. Natarajan, "OCR-Driven Writer Identification and Adaptation in an HMM Handwriting Recognition System," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 739-743, 2011.

[14] F. Perronnin, J. Sanchez and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, pp. 143-156, 2010.

[15] R. Jain and D. Doermann, "Offline writer identification using K-adjacent segments," in *International Conference on Document Analysis and Recognition*

(ICDAR), pp. 769-773, 2011.

[16] R. Jain and D. Doermann, "Writer Identification Using an Alphabet of Contour Gradient Descriptors," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 550-554, 2013.

[17] G. Louloudis, B. Gatos, N. Stamatopoulos and A. Papandreou, "ICDAR 2013 Competition on Writer Identification," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1397-1401, 2013.

[18] R. Jain and D. Doermann, "Combining Local Features for Writer Identification," *International Conference on Frontiers in Handwriting Recognition,* pp. 583 - 588, 2014.

[19] "Tesseract Open Source OCR," [Online]. Available: http://code.google.com/p/tesseract-ocr.

[20] R. Smith, "An overview of the Tesseract OCR engine," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 629-633, 2007.

[21] R. Smith, D. Antonova and D. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," in *Workshop on Multilingual OCR*, 2009.

[22] R. W. Smith, "Hybrid page layout analysis via tab-stop detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 241-245, 2009.

[23] D. W. Oard, B. Hedin, S. Tomlinson and J. R. Baron, "Overview of the TREC 2008 legal track," 2008.

[24] V. Govindaraju and S. Setlur, Guide to OCR for Indic Scripts: Document Recognition and Retrieval, Springer, 2009.

[25] R. Prasad, S. Saleem, M. Kamali, R. Meermeier and P. Natarajan, "Improvements in hidden Markov model based Arabic OCR," in *International Conference on Pattern Recognition (ICPR)*, pp. 1-4, 2008.

[26] S. Saleem, H. Cao, K. Subramanian, M. Kamali, R. Prasad and P. Natarajan, "Improvements in BBN's HMM-based offline Arabic handwriting recognition system," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 773-777, 2009.

[27] D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision and Image Understanding,* vol. 70, no. 3, pp. 287-298, 1998.

[28] S. M. Beitzel, E. C. Jensen and D. A. Grossman, "Retrieving OCR text: A survey of current approaches," in *Symposium on Document Image Understanding Technologies, SDUIT*, 2003.

[29] T. M. Breuel, "The OCRopus open source OCR system.," *IS&T/SPIE Electronic Imaging. Document Recognition and Retrieval,* vol. 6815, p. 68150, 2008.

[30] O. Kolak, W. Byrne and P. Resnik, "A generative probabilistic OCR model for NLP applications," in *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 55-62, 2003.

[31] K. Taghva, J. Borsack, A. Condit and S. Erva, "The Effects of noisy data on text retrieval," *Journal of the ASIS,* vol. 45, no. 1, pp. 50-58, 1994.

[32] K. Taghva, J. Borsack and A. Condit, "Effects of OCR errors on ranking and feedback using the vector space model," *Information processing & management,* vol. 32, no. 3, pp. 317-327, 1996.

[33] S. M. Harding, W. B. Croft and C. Weir, "Probabilistic retrieval of ocr degraded text using n-grams," in *Research and advanced technology for digital libraries*, Springer, 1997, pp. 345-359.

[34] R. F. Bulcao-Neto, J. A. Camacho-Guerrero, M. Dutra, A. Barreiro, J. Parapar and A. A. Macedo, "The Use of Latent Semantic Indexing to Mitigate OCR Effects of Related Document Images," *Journal of Universal Computer Science,* vol. 17, no. 1, pp. 64-80, 2011.

[35] E. Hassan, V. Garg, S. M. Haque, S. Chaudhury and M. Gopal, "Searching OCR'ed Text: An LDA Based Approach," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1210-1214, 2011.

[36] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1162-1173, 1993.

[37] K. Kise, A. Sato and M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Computer Vision and Image Understanding,* vol. 70, no. 3, pp. 370-382, 1998.

[38] M. Agrawal and D. Doermann, "Voronoi++: A dynamic page segmentation approach based on voronoi and docstrum features," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1011-1015, 2009.

[39] S. Mao, A. Rosenfeld and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Electronic Imaging 2003*, pp. 197-207, 2003.

[40] C. Shin, D. Doermann and A. Rosenfeld, "Classification of document pages using structure-based features," *International Journal on Document Analysis and Recognition,* vol. 3, no. 4, pp. 232-247, 2001.

[41] M. Huang, D. DeMenthon, D. Doermann, L. Golebiowski and B. A. Hamilton, "Document ranking by layout relevance," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 362-366, 2005.

[42] A. Gordo and E. Valveny, "A rotation invariant page layout descriptor for document classification and retrieval," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 481-485, 2009.

[43] S. Marinai, E. Marino and G. Soda, "Tree clustering for layout-based document image retrieval," in *International Conference on Document Image Analysis for Libraries*, pp. 243-253, 2006.

[44] T. Nakai, K. Kise and M. Iwamura, "Hashing with local combinations of feature points and its application to camera-based document image retrieval," *International Workshop on Camera-Based Document Analysis and Recognition,* pp. 87-94, 2005.

[45] K. Takeda, K. Kise and M. Iwamura, "Real-time document image retrieval for a

10 million pages database with a memory efficient and stability improved LLAH," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1054-1058, 2011.

[46] D. Doermann, E. Rivlin and I. Weiss, "Applying algebraic and differential invariants for logo recognition," *Machine Vision and Applications,* vol. 9, no. 2, pp. 73-86, 1996.

[47] G. Zhu and D. Doermann, "Automatic document logo detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 864-868, 2007.

[48] H. Wang and Y. Chen, "Logo detection in document images based on boundary extension of feature rectangles," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1335-1339, 2009.

[49] Z. Li, M. Schulte-Austum and M. Neschen, "Fast logo detection and recognition in document images," in *International Conference on Pattern Recognition (ICPR)*, pp. 2716-2719, 2010.

[50] M. Rusinol and J. Llados, "Efficient logo retrieval through hashing shape context descriptors," in *International Workshop on Document Analysis Systems*, pp. 215-222, 2010.

[51] G. Zhu and D. Doermann, "Logo matching for document image retrieval," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 606-610, 2009.

[52] M. Rusinol, D. Aldavert, R. Toledo and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 63-67, 2011.

[53] M. J. Fonseca, A. Ferreira and J. A. Jorge, "Content-based retrieval of technical drawings," *International Journal of Computer Applications in Technology,* vol. 23, no. 2, pp. 86-100, 2005.

[54] D. Impedovo and G. Pirlo, "Automatic signature verification: the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 38, no. 5, pp. 609-635, 2008.

[55] S. N. Srihari, S. Shetty, S. Chen, H. Srinivasan, C. Huang, G. Agam and O. Frieder, "Document image retrieval using signatures as queries," in *International Conference on Document Image Analysis for Libraries*, pp. 198-203, 2006.

[56] G. Agam, S. Argamon, O. Frieder, D. Grossman and D. Lewis, "Content-based document image retrieval in complex document collections," in *Electronic Imaging 2007*, p. 65000S, 2007.

[57] G. Zhu, Y. Zheng, D. Doermann and S. Jaeger, "Multi-scale structural saliency for signature detection," in *Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[58] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal of Document Analysis and Recognition (IJDAR),* vol. 9,

no. 2-4, pp. 139-152, 2007.

[59] M. Rusinol and J. Llados, "Logo spotting by a bag-of-words approach for document categorization," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 111-115, 2009.

[60] Y. Ke, R. Sukthankar and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *ACM Multimedia*, pp. 869-876, 2004.

[61] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," *European Conference on Computer Vision,* pp. 404-417, 2006.

[62] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[63] A. Auclair, N. Vincent and L. D. Cohen, "Hash functions for near duplicate image retrieval," in *Workshop on Applications of Computer Vision (WACV)*, pp. 1-6, 2009.

[64] O. Chum, J. Philbin and A. Zisserman, "Near Duplicate Image Detection: min-Hash and tf-idf Weighting.," in *British Machine Vision Conference*, pp. 812-815, 2008.

[65] M. D. G. L. Muja, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *International Conference on Computer Vision Theory and Applications*, pp. 331-340, 2009.

[66] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, no. 6, pp. 381-395, 1981.

[67] "The Legacy Tobacco Document Library (LTDL)," University of California, San Francisco, 2007. [Online]. Available: http://legacy.library.ucsf.edu.

[68] S. Tomlinson, D. W. Oard, J. R. Baron and P. Thompson, "Overview of the TREC 2007 Legal Track," in *TREC*, 2007.

[69] "The Lucene IR Engine," [Online]. Available: http://lucene.apache.org/.

[70] C. Buckley and E. M. Voorhees, "Retrieval evaluation with incomplete information," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 25-32, 2004.

[71] I. Staff, "Measuring and delivering 95% non-stopword document accuracy," Technical Report 2003-04, Information Science Research Institute, 2003.

[72] D. Carmel and E. Yom-Tov, "Estimating the query difficulty for information retrieval.," *Synthesis Lectures on Information Concepts, Retrieval, and Services,* vol. 2, no. 1, pp. 1-89, 2010.

[73] The National Archives, "Franklin," 2014. [Online]. Available: http://www.fdrlibrary.marist.edu/archives/collections/franklin/.

[74] D. Doermann, H. Li and O. Kia, "The detection of duplicates in document image databases," *Image and vision computing,* vol. 16, no. 12, pp. 907-920, 1998.

[75] J. J. Hull, "Document image similarity and equivalence detection," *International Journal on Document Analysis and Recognition,* vol. 1, no. 1, pp.

37-42, 1998.

[76] D. P. Lopresti, "A comparison of text-based methods for detecting duplication in scanned document databases," *Information Retrieval,* vol. 4, no. 2, pp. 153-173, 2001.

[77] M. Jiang, E. K. Wong and N. Memon, "Robust document image authentication," in *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 1131-1134, 2007.

[78] E. W. Myers, "An O (ND) difference algorithm and its variations," *Algorithmica,* vol. 1, no. 1-4, pp. 251-266, 1986.

[79] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence,* vol. 17, no. 1, pp. 185-203, 1981.

[80] Y. Sankarasubramaniam, B. Narayanan, K. Viswanathan and A. Kuchibhotla, "Detecting modifications in paper documents: a coding approach," in *IS&T/SPIE Electronic Imaging*, 2010.

[81] P. Clough, R. Gaizauskas, S. Piao and Y. Wilks, "METER: MEasuring TExt Reuse," *Annual Meeting on Association for Computational Linguistics,* pp. 152-159, 2002.

[82] A. Sayeed, S. Sarkar, Y. Deng, R. Hosn, R. Mahindru and N. Rajamani, "Characteristics of Document Similarity Measures for Compliance Analysis," *ACM conference on Information and knowledge management,* pp. 1207-1216, 2009.

[83] A. Rush, "Redaction Analysis," 2013. [Online]. Available: http://declassification-engine.org/redactions.

[84] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, vol. 2, Cambridge Univ Press, 2000.

[85] A. Bosch, A. Zisserman and X. Munoz, "Scene classification via pLSA," in *European Conference on Computer Vision*, pp. 517-530, 2006.

[86] C. Silpa-Anan, and R. Hartley, "Optimised KD-trees for fast image descriptor matching," *CVPR,* pp. 1-8, 2008.

[87] S. Russell and P. Norvig, Artificial Intelligence. "A modern approach", 3rd edition ed., Prentice-Hall, 2009, p. 125.

[88] S. N. Srihari, S.-H. Cha, H. Arora and S. Lee, "Individuality of handwriting: a validation study," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 106-109, 2001.

[89] A. Schlapbach and H. Bunke, "A writer identification and verification system using HMM based recognizers," *Pattern Analysis and Applications,* vol. 10, no. 1, pp. 33-43, 2007.

[90] A. Schlapbach and H. Bunke, "Off-linewriter identification using Gaussian mixture models," in *International Conference on Pattern Recognition (ICPR)*, pp. 992-995, 2006.

[91] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, pp. 701-717, 2007.

[92] S. Fiel and R. Sablatnig, "Writer Retrieval and Writer Identification using Local Features," in *International Workshop on Document Analysis Systems*, pp. 145-149, 2012.

[93] L. Schomaker, M. Bulacu and K. Franke, "Automatic writer identification using fragmented connected-component contours," in *International Workshop on Frontiers in Handwriting Recognition*, pp. 185-190, 2004.

[94] M. Bulacu, L. Schomaker and A. Brink, "Text-independent writer identification and verification on offline Arabic handwriting," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 769-773, 2007.

[95] M. N. Abdi, M. Khemakhem and H. Ben-Abdallah, "A novel approach for off-line Arabic writer identification based on stroke feature combination," in *International Symposium on Computer and Information Sciences*, pp. 597-600, 2009.

[96] Z. He, X. You and Y. Y. Tang, "Writer identification of Chinese handwriting documents using hidden Markov tree model," *Pattern Recognition,* vol. 41, no. 4, pp. 1295-1307, 2008.

[97] V. Ferrari, L. Fevrier, F. Jurie and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 1, pp. 36-51, 2008.

[98] G. Zhu, X. Yu, Y. Li and D. Doermann, "Language identification for handwritten document images using a shape codebook," *Pattern Recognition,* vol. 42, no. 12, pp. 3184-3191, 2009.

[99] J. Kumar, R. Prasad, H. Cao, W. Abd-Almageed, D. Doermann and P. Natarajan, "Shape codebook based handwritten and machine printed text zone extraction," in *IS&T/SPIE Electronic Imaging*, pp. 787406-787406, 2011.

[100] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science,* vol. 315, no. 5814, pp. 972-976, 2007.

[101] C. I. Tomai, B. Zhang and S. N. Srihari, "Discriminatory power of handwritten words for writer recognition," in *International Conference on Pattern Recognition (ICPR)*, pp. 638-641, 2004.

[102] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM Transactions on graphics (TOG)*, p. 10, 2007.

[103] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition,* vol. 5, no. 1, pp. 39-46, 2002.

[104] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, pp. 701-717, 2007.

[105] X. Wu, Y. Tang and W. Bu, "Offline Text-independent Writer Identification Based on Scale Invariant Feature Transform," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 3, pp. 526-536, 2013.

[106] S. Fiel and R. Sablatnig, "Writer Identification and Writer Retrieval using the Fisher Vector on Visual Vocabularies," *International Conference on Document*

Analysis and Recognition (ICDAR), pp. 545-549, 2013.

[107] C. Djeddi, I. Siddiqi, L. Souici-Meslati and A. Ennaji, "Text-independent writer recognition using multi-script handwritten texts," *Pattern Recognition Letters,* vol. 34, no. 10, pp. 1196-1202, 2013.

[108] J. Woodard, M. Lancaster, A. Kundu, D. Ruiz and J. Ryan, "Writer recognition of Arabic text by generative local features," *Biometrics: Theory Applications and Systems (BTAS),* pp. 1-7, 2010.

[109] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech communication,* vol. 52, no. 1, pp. 12-40, 2010.

[110] I. Siddiqi and N. Vincent, "Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features," *Pattern Recognition,* vol. 43, no. 11, pp. 3853-3865, 2010.

[111] F. Kleber, S. Fiel, M. Diem and R. Sablatnig, "CVL-Database: An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting," *International Conference on Document Analysis and Recognition (ICDAR),* pp. 560-564, 2013.

[112] X. Li and X. Ding, "Writer identification of Chinese handwriting using grid microstructure feature," in *Advances in Biometrics*, pp. 1230-1239, 2009.

[113] S. Strassel, "Linguistic resources for Arabic handwriting recognition," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt*, 2009.

[114] J. Chen, D. Lopresti and E. Kavallieratou, "The impact of ruling lines on writer identification," in *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pp. 439-444, 2010.

[115] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[116] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional," *Advances in neural information processing systems,* pp. 1097-1105, 2012.

[117] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *Computer Vision and Pattern Recognition (CVPR),* pp. 1701-1708, 2014.

[118] D. Harrison, T. M. Burkes and D. P. Seiger, "Handwriting examination: meeting the challenges of science and the law," *Forensic Science Communications,* vol. 11, no. 4, 2009.

[119] S. M. Awaida and S. A. Mahmoud, "State of the art in off-line writer identification of handwritten text and survey of writer identification of Arabic text," *Educational Research and Reviews,* vol. 7, no. 20, pp. 445-463, 2012.

[120] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant and others, "Quantitative analysis of culture using millions of digitized books," *science,* vol. 331, no. 6014, pp. 176-182, 2011.

[121] R. A. Huber and A. M. Headrick, Handwriting identification: facts and fundamentals, CRC Press, 2010.

[122] N. Cheng, G. K. Lee, B. S. Yap, L. T. Lee, S. K. Tan and K. P. Tan, "Investigation of Class Characteristics in English Handwriting of the Three Main Racial Groups: Chinese, Malay, and Indian in Singapore," *J Forensic Sci,* vol. 50, no. 1, pp. 1-8, 2005.

[123] A. Hassaine, S. Al Maadeed, J. Aljaam and A. Jaoua, "ICDAR 2013 Competition on Gender Prediction from Handwriting," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1417-1421, 2013.

[124] L. Xu, X. Ding, L. Peng and X. Li, "An improved method based on weighted grid micro-structure feature for text-independent writer recognition," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 638-642, 2011.

[125] I. Siddiqi and N. Vincent, "A set of chain code based features for writer recognition," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 981-985, 2009.

[126] Y. Li, Y. Zheng, D. Doermann and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 8, pp. 1313-1329, 2008.

[127] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 63-84, 2000.

[128] G. Zhu, Y. Zheng, D. Doermann and S. Jaeger, "Signature detection and matching for document image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, no. 11, pp. 2015-2031, 2009.

[129] R. J. Radke, S. Andra, O. Al-Kofahi and B. Roysam, "Image change detection algorithms: a systematic survey," *IEEE Transactions on Image Processing,* vol. 14, no. 3, pp. 294-307, 2005.

[130] G. Louloudis, B. Gatos and N. Stamatopoulos, "ICFHR 2012 Competition on Writer Identification Challenge 1: Latin/Greek Documents," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 829-834, 2012.