

ATHABASCA UNIVERSITY

SECURING TRANSACTION IMAGE FILES USING DIGITAL
WATERMARKING

BY

RONALD EDEN BURTON

A thesis submitted in partial fulfillment
Of the requirements for the degree of
MASTER OF SCIENCE in INFORMATION SYSTEMS

Athabasca, Alberta

November, 2006

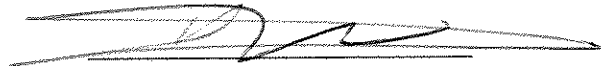
© Eden Burton, 2006

ATHABASCA UNIVERSITY

The undersigned certify that they have read and recommend for acceptance the project
“SECURING TRANSACTION IMAGE FILES USING DIGITAL WATERMARKING”
submitted by RONALD EDEN BURTON in partial fulfillment of the requirements for
the degree of MASTER OF SCIENCE in INFORMATION SYSTEMS.



Xiaokun Zhang, PhD
Supervisor



Eric (Yuefei) Xu, PhD
Examiner



Mahmoud Abaza, PhD
Chair

Date: January 2017

ABSTRACT

This research paper examines the problem of guaranteeing electronic document integrity. Digital watermarking will be presented as a potential means of accomplishing this goal. The paper proposes an electronic exam exchange program as a frame of reference for the research conducted.

In the case of an exam document exchange system, one must accommodate the transfer of both text and handwritten information within these documents. Exams are typically in ASCII text format and printed onto paper so that students can handwrite answers in the appropriate location. The system must be constructed in a way so that security measures are not defeated when the document is transformed from paper to electronic format. The research paper will present a potential method for dealing with this issue.

During a document's existence in paper form, its quality can be degraded significantly. Paper can be folded, soiled or marked. During the rescanning of this document, the document can be degraded to the point where its security markings can become unusable. After scanning, common operations on an image file such as cropping, rotation and compression can also negatively impact its security marks.

This paper proposes a watermarking technique that uses optical character recognition to verify the integrity of security artifacts embedded in a digital file. The experiment outlined in the paper was conducted using a prototype modelling the technique described. The results of this experiment are used to evaluate the technique's utility.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the faculty at Athabasca's School of Computing for preparing me for the challenge of producing my first published work.

Special thanks go out to Dr. Zhang for accepting the task of supervising this project.

Finally, I would like to acknowledge Janice Green and the Athabasca University Graduate Student Research Fund Committee for providing the equipment used to perform this project's experiment.

TABLE OF CONTENTS

CHAPTER I - INTRODUCTION	1
Statement of Purpose.....	1
Research Question or Purpose	1
Significance	5
Organization of the Project.....	6
CHAPTER II – REVIEW OF RELATED LITERATURE.....	8
Digital Image Processing.....	8
OCR.....	19
Security.....	29
Digital Watermarking.....	30
Attributes of Watermarking Schemes	30
Watermarking Defined	33
Watermarking Techniques.....	40
Increasing the Robustness of Watermarks.....	44
Watermark Security.....	46
Watermarking Attacks.....	47
CHAPTER III - METHODOLOGY	50
Proposed Solution Design	50
Exam Document.....	52
Watermark Experiment	54
Overview.....	54
Objective.....	55

Tools	55
AU Watermarking Application	56
Procedure	57
Data Analysis Techniques	60
CHAPTER IV – RESULTS	62
Technical Considerations	62
Commercial Feasibility Considerations	66
CHAPTER V – CONCLUSIONS AND RECOMMENDATIONS	69
REFERENCES	73
APPENDIX A	78
APPENDIX B	79
APPENDIX C	85
APPENDIX D	88

LIST OF TABLES

Table 1: Fields for Exam Watermark.....	50
Table 2: Outline Of The Proposed Exam Watermarking System.....	51
Table 3: Results of Running Character Recognition On Extracted Watermarks	63
Table 4: Results From Extracting Watermarks From Cover Works.....	65

LIST OF FIGURES

Figure 1: Current Logical Exam Exchange System at Athabasca University.....	3
Figure 2: A Digital Representation of an Image.....	9
Figure 3: An Example of Differences in Sampling Resolution.....	10
Figure 4: Image Filtering Models	13
Figure 5 – from Image Processing Handbook (J. Ross).....	15
Figure 6: The 2-D Fourier Transform	16
Figure 7: The 2-D Inverse Fourier Transform.....	16
Figure 8: Discrete Version of the Fourier Transform	17
Figure 9: 2-D Wavelet Transform.....	18
Figure 10: Mother Wavelet Equation.....	18
Figure 11: An Example of Text Hidden in a Picture	21
Figure 12: Parker’s Method Outline For Locating Text on White Paper.....	23
Figure 13: Baird’s Method For Skew Detection In Text	25
Figure 14: Using Horizontal Projections of Text for Skew Correction	26
Figure 15: Mathematical Representation Of The Watermarking Embedding Process.....	33
Figure 16: Mathematical Representation Of The Watermarking Detection Process.....	33
Figure 17: Mathematical Representation Of The Watermark Extraction Process	34
Figure 18: A Generic Watermarking System Design	35
Figure 19: A Geometric View of the Watermark Image Space.....	37
Figure 20: Division of the Image Space Into Disjointed Blocks For Watermarking Purposes.....	40
Figure 21: A Sample Exam Document	53

Figure 22: File Size of Watermarks Created by the AU Watermarking System.....	64
Figure 23: Time Required To Obtain a Letter Size Image From The Scanner	67
Figure 24: File Size of Cover Works Captured By The Scanner.....	68
Figure 25: Projection Equation.....	79
Figure 26: Equation For Cross-Correlation.....	80
Figure 27: Recognition Based On Boolean Algebra.....	82
Figure 28: An Example Of Weeks's Recognition Analysis.....	84

CHAPTER I

INTRODUCTION

Statement of Purpose

The purpose of this research paper is to determine if document integrity can be guaranteed using current digital watermarking techniques. The documents in question originate in electronic form but temporarily exist on physical paper so that handwritten information can be inserted. Exam document management at Athabasca University (AU) will be the focus of this study.

Research Question or Purpose

In today's society, many transactions between individuals and organizations alike are conducted via electronic means. By exchanging transaction details digitally, parties can reduce or eliminate costs associated with tracking and storing paper. Transactions represented electronically are typically housed in a data store for future reference.

Electronic based transaction systems offer the advantages of being fast, cost effective and convenient for processing transactions that do not require face-to-face contact. These advantages are usually realized over public communication channels. Guaranteeing a suitable level of security for transaction data in transit is critical for a commercial quality transaction management system.

At academic institutions, transactions between faculty and students occur during final evaluation processes. A professor gives students a set of questions to answer within a

given time period. In turn, students provide answers to these questions for grading.

Distance education providers such as Athabasca University use independent proctoring services to facilitate this type of transaction.

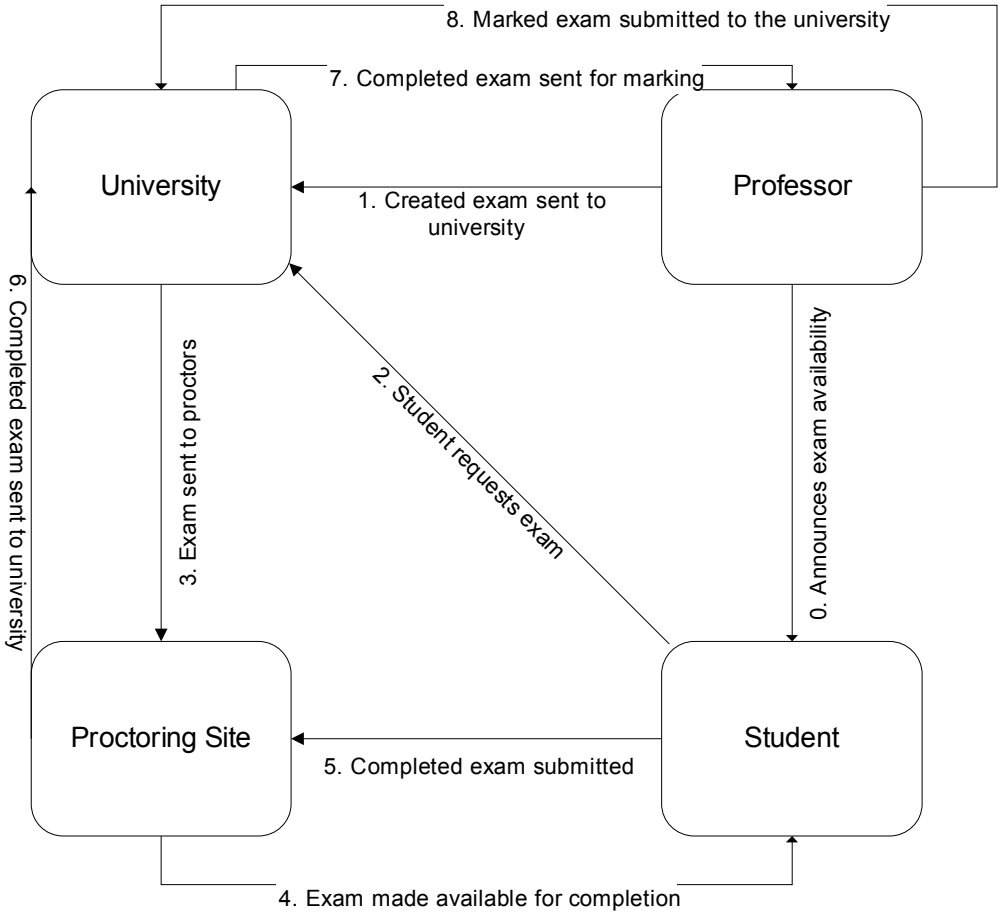
Currently, students at Athabasca University request final exams using a form provided with their course materials. Under normal circumstances, the school requires 15 days notice so that it can guarantee that exams arrive at the appropriate proctoring sites on time. On the request form, students choose the location they would like to write their exam. Athabasca University has established proctoring agreements with a large group of public and private institutions. These centers provide weekly sessions that Athabasca students can use to write their exams.

Upon arrival at the institution on exam day, a student fills out a confirmation form, pays the sitting fee and writes the exam. When the examination is complete, the proctor signs the student's confirmation form, collects the written exam and mails both documents to the University for processing.

Athabasca University uses registered mail or courier to exchange exam documents with the proctoring institutions. This form of delivery is used because it requires positive acknowledgement of receipt and as such, it is unlikely that the documents would become lost or read by undesired parties. The exams are stored in the University under lock and key. At the proctoring institutions, exam storage is not standardized; however, they might be bound to agreements ensuring that the exams are held in a secure place while in their

possession. Figure 1 illustrates how Athabasca University’s final exam process currently works.

Figure 1: Current Logical Exam Exchange System at Athabasca University



Exam documents passed between the involved parties exist in paper format. If these exam documents are converted into electronic format, they can be exchanged between the University and its proctoring site via the Internet. This would eliminate postage fees required to exchange the exams. In addition, the time required to transport the exams

between parties would be negligible compared to shipping paper between different locations.

A challenge encountered when transmitting highly confidential documents is ensuring that the security of embedded information is not compromised. The context of the exam exchange system currently being examined has two critical features that must be maintained in any electronic system proposed to replace it: confidentiality and integrity.

Athabasca University's current exam exchange system ensures that confidentiality of the exams is maintained by sending them using a trusted mail or courier system. It is assumed that the trusted party used does not allow any unauthorized persons to open mail pieces containing exams while in its custody. Furthermore, the trusted party guarantees that exams are delivered to their destination within an agreed timeframe.

In the paper-based system currently in use, sealed packaging ensures the integrity of the exam documents while in transit between physical locations. The University has agreements in place with its employees and proctoring institutions to ensure the integrity of these documents at other points in the process. For example, a student writing an exam at a proctoring center is not allowed to leave the designated examination sitting room with any pieces of paper as they may contain confidential exam data.

This report will evaluate the feasibility of using digital watermarking as a means of guaranteeing integrity of exam documents within the exam exchange system being

discussed. Digital watermarking techniques embed information in the actual computer file being protected. This method is commonly used to enable copyright protection in multimedia applications. For digital watermarking to be used successfully in the exam exchange system, it must survive a document's conversion from digital to paper format. After a student or evaluator adds handwritten information onto an exam paper, it is scanned and converted back into digital format. During a document's existence in paper form, its quality can be degraded significantly. Paper can be folded, soiled or marked. During the rescanning of this document, it can be degraded to the point where its security watermarks can become unusable. After scanning, common operations on an image file such as cropping, rotation and compression can also negatively impact its security marks. A prototype will be developed to evaluate the utility of digital watermarking for providing exam document integrity within the Athabasca University exam system.

Significance

The advantages of a computerized, distributed transaction management system over a paper based one have been documented in the previous section. A computerized solution meeting these requirements will allow Athabasca University to reduce costs while potentially improving quality of service. These advantages cannot be realized in a production system if the solution does not have a suitable level of security to protect its transactions.

If a suitable security mechanism for exam images can be found, it can also be used to develop transaction management solutions for other types of applications. For example, a health care system that requires the exchange of confidential patient and doctor records would require similar levels of security to that of the university exam system being discussed. Other examples can be found in both the legal and financial industries.

Organization of the Project

The first section of this paper is meant to give the reader a description of the Athabasca University exam exchange system being evaluated. It describes features that are required in such a system. It also states the goal of the researcher.

The literary review chapter gives the reader an introduction to subject matter areas that are to be utilized in the design of an electronic exam exchange system. Current practice and basic theory in the respective areas are outlined.

The design of a proposed alternate exam exchange system will be described in the methodology section of this paper. A prototype will be implemented in order to determine the feasibility of securing documents which are converted between digital and hard copy formats. A particular marking technique selected by the author will attempt to accomplish this. The selected technique will be evaluated based on the requirements of the system being modeled.

Data collected by the testing tool will be summarized in the results section. An assessment of the data will be conducted and any findings will be articulated at this point.

The paper will conclude with a section dedicated to interpreting the study's findings and commenting on the feasibility of using the selected technique to implement the proposed exam exchange system. Suggested directions for future research complete the discussion in this section of the paper.

CHAPTER II

REVIEW OF RELATED LITERATURE

Digital Image Processing

Image processing tools take images and alter them in a way which makes them more useful to an end-user. The goal for such tools might be to make images more aesthetically pleasing or it may be to make them easier to extract information from. Performing these operations in the digital domain allow complex, non-destructive manipulation of images in an efficient manner.

A digital image is a two dimensional array of numeric values (Niblack, 1986). Each value represents a discrete point in an image. These points are typically organized in a grid and identified as shown in Figure 2. Images of high quality are sampled using many points or pixels per image unit area. Baxes (1984) states in his work that the density, or resolution, of a digital image should be set such that two pixels lie where any variance in brightness can be detected. This high density of points will accurately capture a real world scene and eliminate the possibility of the human eye detecting any difference between the digital representation and the original scene. Practical considerations often prevent this level of sampling from being possible however sampling rates below these levels can still obtain satisfactory results. Figure 3 displays two images of the same scene captured at different sampling rates. These images allow one to appreciate how a capture device's sampling rate affects image quality.

Figure 2: A Digital Representation of an Image

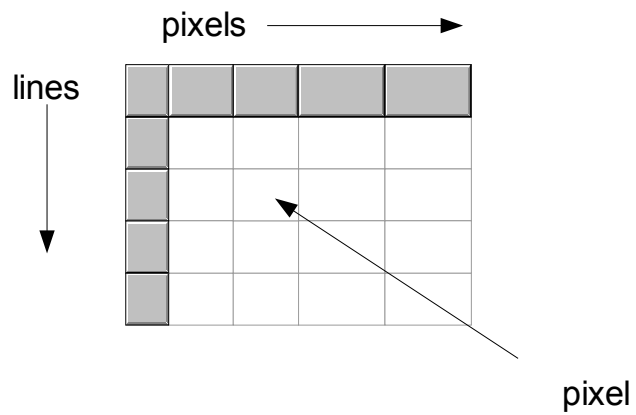


Figure 3: An Example of Differences in Sampling Resolution



Sampling Rate: 1444 pixels/sq. cm



Sampling Rate: 25 pixels/sq. cm

Each point's numeric value is used to tell the rendering device (i.e. computer monitor or printer) how it should be drawn on the output medium. In achromatic, or "black and white", images the value represents a shade of grey or a level of brightness. In colour images, the number represents an actual colour. There are a few common colour models in use today. In systems rendering graphics to a computer monitor, the RGB system is typically used. In this system, the numeric value assigned specifies the amount of red, green and blue energy to include in the pixel's output. The various combinations of these primary colours allow the whole spectrum of possible colours to be displayed. The presence of none of the primaries produces the colour black. Alternatively, all the primaries combined at maximum energy produces white. In systems defining output for printing devices, a model was developed that "subtracted" from white (the most popular colour for printer paper) as opposed to "added" to black. It turns out that the definition of three new primaries, cyan, magenta and yellow, accomplishes this goal and the CMY model based on them is complimentary to the RGB colour model. The YIQ colour model is used in colour broadcasting. It effectively separates luminance (stored in the component Y), which can loosely be defined as brightness, from chromaticity information (stored in the components I and Q). This provides value as the human eye "is more sensitive to changes in luminance than to changes in hue or saturation". (Foley, van Dam, Feiner, Hughes & Phillips, 1994, p. 413) Rendering systems can therefore maintain more detailed information about Y at the expense of the other components without much loss of perceived quality. Conversion methods are available to convert information between models. Foley, Van Dam et al. (1994) dedicate a complete chapter of their work to colour models and should be referenced for further details.

Storing images in the digital domain gives the end user the ability to perform a wide variety of operations instantaneously. Operations that alter the brightness or colour of an image can be performed by modifying its pixel information using the colour models previously mentioned. Images or parts thereof can also be stretched, moved or rotated with the application of simple algebraic functions. Appendix A lists examples of some basic geometric operations that can be done. Multiple operations can be performed concurrently by combining the appropriate equations.

End users are often interested in the spatial frequency information found in an image. “An image is said to be composed of many basic frequency subcomponents, ranging from high to low” (Baxes, 1984, p. 47). High spatial frequency occurs in an image where rapid changes in brightness reside. Alternatively, low frequencies occur in portions of an image where brightness is relatively constant. Frequency information is of interest because the human vision system is sensitive to changes in brightness. It uses high frequencies to identify objects and separate areas of an image. Spatial filtering can be used to accentuate either high or low frequency information in an image. This process is done by having a pixel’s brightness assigned to be the weighted average of brightness values from some set of neighbouring pixels. The weighting and neighbouring pixel set size is defined to achieve a particular effect. Smoothing or low pass filters are used to reduce noise or detail in an image. They tend to set a pixel’s brightness close to the value of its neighbours. A significant challenge is to smooth out areas without blurring details of interest. Edge enhancement filters attempt to magnify the differences in brightness

between neighbouring pixels. Figure 4 shows two examples of spatial filters. In Figure 4 a), the filter defined will have the highlighted pixel's luminance set to an average of its luminance and all of its neighbouring pixels. When applied to all pixels in an image, the process smoothes out edges contained within it. Using the same concept, Figure 4 b) implements an edge enhancement filter. It does so by setting the luminance of each pixel to five times its own luminance minus the luminance of its vertical and horizontal neighbours.

Figure 4: Image Filtering Models

	1/9	1/9	1/9	
	1/9	1/9	1/9	
	1/9	1/9	1/9	

a) Edge smoothing, square shaped window

		-1		
	-1	5	-1	
		-1		

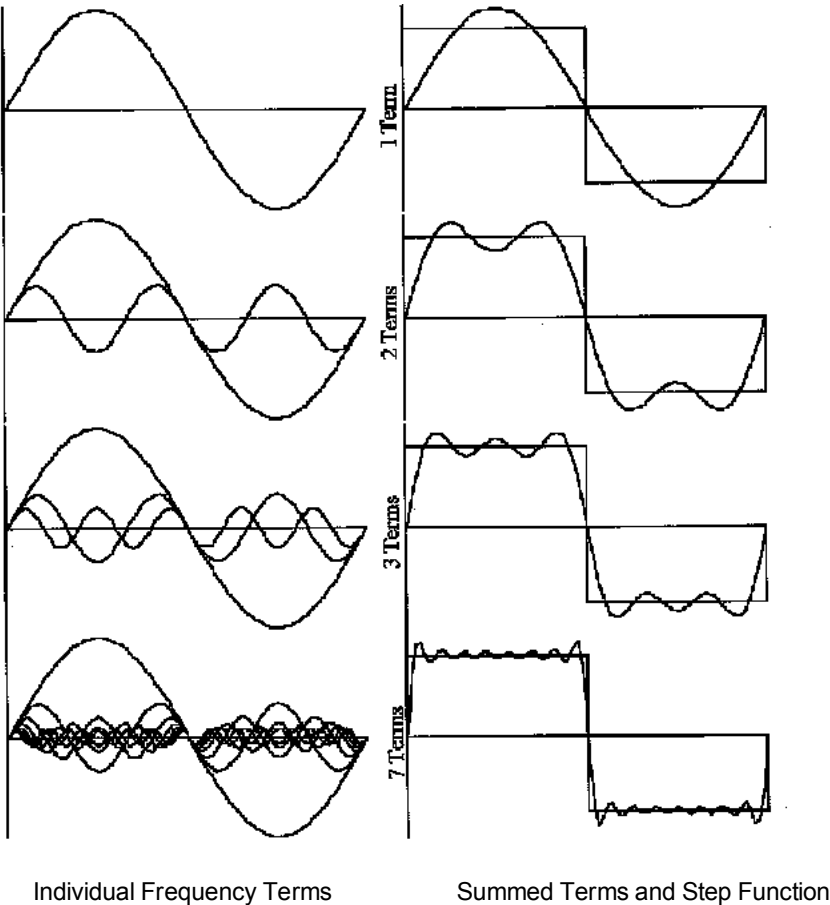
b) Edge enhancement, star shaped window

So far, the image manipulation methods presented have been done in the spatial domain. Some image processing and analysis however is simpler to do in an alternate system. By transforming an image into a different space, different attributes come to the forefront. In this section, examples of such methods are discussed. The Fourier and wavelet based

transforms move an image into a space that is appropriate for spatial frequency operations. The Hough transform moves images into a representation that exposes straight lines. Each will now be discussed in turn.

The Fourier transform provides a useful representation for isolating the various levels of spatial frequencies in an image. Fourier's theorem states that any image can be expressed by the summation of sine and cosine waves. The waves are of varying frequencies and amplitudes. Figure 5 derived from Russ's work (1992) shows how a simple step function in one dimension is approximated by adding sincoid waves together. The low frequency waves provide the 'body' of the function while the higher frequency terms add detail to the approximation. This is consistent with our discussion about spatial frequency in the previous section. The transform moves the image from a domain where the image is expressed as brightness as a function of spatial placement to where it is a set of amplitudes corresponding to frequencies of sincoid and cosine waves. The Fourier transform is stated in Figure 6 below.

Figure 5 – from Image Processing Handbook (J. Ross)



Individual Frequency Terms

Summed Terms and Step Function

Figure 6: The 2-D Fourier Transform

$$F(\omega, \nu) = \int \int f(x, y) \ell^{-j2\pi(\omega x + \nu y)} dx dy$$

where $j = \sqrt{-1}$

and $\ell^{-j2\pi(\omega x + \nu y)} = \cos(2\pi\omega x) - j \sin(2\pi\omega x) + \cos(2\pi\nu y) - j \sin(2\pi\nu y)$

There is a unique one to one relationship between $f(\omega, \nu)$ and $f(x, y)$. There is an inverse function that transforms an image from the frequency domain into the spatial domain. This function is listed in Figure 7. Transforming between the spatial and frequency domain (and vice-versa) is done without any loss of image data.

Figure 7: The 2-D Inverse Fourier Transform

$$f(x, y) = \int \int F(\omega, \nu) \ell^{j2\pi(\omega x + \nu y)} d\omega d\nu$$

In practice, this equation is not used because of the discrete nature of digital images and the fact that spatial frequencies do not get higher than the Nyquist frequency. The discrete version of the transform is mentioned in Figure 8.

Figure 8: Discrete Version of the Fourier Transform

$$F(h, i) = \frac{1}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} f(k, l) \ell^{-j 2\pi(kh+li)/n}$$

where $j = \sqrt{-1}$

and $0 \leq h, i \leq n - 1$

Low-pass and high-pass filtering shown in the previous section is achieved by zeroing out certain terms of the transform. Eliminating a particular term removes the corresponding frequency level throughout the image. Therefore, filtering can be done by performing a Fourier transform, zeroing out the desired frequency level and calculating the inverse transform to return the altered image back into the spatial domain.

One weakness of the Fourier transform is that it decomposes an image into its frequency components but its representation provides no information about where in the image these frequencies occur. Unfortunately, it is impossible to obtain the spatial frequency in an image at a particular location due to Heisenberg's uncertainty principle. Wavelet transforms attempt to overcome this limitation by expressing an image as a linear combination of different scalings and translations of some defined wavelet function known as the mother wavelet. This wavelet function must have periodicity and its value must decrease as some function of the distance from its centre. Note that the transform does not specify exactly what the mother wavelet function should be. In practice, the

function is customized to suit the particular application. The following equation in Figure 9 states the one-dimensional wavelet transform. In two-dimensional functions, such as images, “the two-dimensional version of a wavelet transform can be expressed in terms of a number of one-dimensional transforms.” (Parker, 1997, p. 269). Wavelet transforms are often used for applications such as image filtering and compression.

Figure 9: 2-D Wavelet Transform

$$\gamma_{f(t)}(s, \tau) = \int f(t) \psi_{s, \tau}(t) dt$$

where the mother wavelet is defined as $\psi(t)$ and the series of wavelet is defined as $\psi(t)_{s, \tau}$ is defined by the equation noted in Figure 10.

Figure 10: Mother Wavelet Equation

$$\psi_{s, \tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right)$$

Hough's transform represents each pixel in the standard spatial representation as a line in what is known as the Hough space. The equation for a line in the standard domain is $y = mx + b$ where m and b are constants. In the Hough space, this equation is rearranged to be $b = -xm + y$ where x and y are constant terms. Lines that intersect in the Hough space

represent pixels in the standard domain that reside on the same line. This provides a novel way of line detection in images.

This section provided the reader with some background knowledge about image processing. This information will be useful when reading rest of this paper.

OCR

Computer applications can write human readable information to paper using a printing device such as a plotter or printer. When a computer is instructed to print a character to an output device, it simply looks up the pre-defined bitmap set used to generate the desired character. A defined set of bitmaps (or an equivalent representation) is stored within the computer system for each character in a language's alphabet. The character lookup is done using the machine's representation of the character. The actual bitmap retrieved within the selected set is based on the visual attributes desired. Once the actual bitmap is found, appropriate instructions are sent to a printing device to render the map on paper. Assuming that the output is printed in a colour with sufficient contrast to the colour of the paper, it can be easily interpreted by humans. Translation in the opposite direction has proven to be a much more difficult problem to solve. Collecting character information from paper is done using a scanning device. This device produces a digital snapshot of the sheet of paper. In order for the computer to extract the required character information, it must simulate the process of a human reading and interpreting the snapshot. The field of optical character recognition (OCR) involves the study of techniques used by a computer to extract character-based information from such

snapshots. This discipline draws from the fields of artificial intelligence, image processing and other areas of computer science in order to define algorithms for accomplishing its goals. The rest of this section will present a general process for performing character recognition, identify problems encountered when attempting it and illustrate approaches proposed by researchers in the field.

Gatos, Papamarkos et al (1997) divide the OCR process into three specific stages: preprocessing, feature extraction and classification.

At the preprocessing stage, the primary goal is to isolate the glyphs or visual character representations from the image of the paper being read. To accomplish this, the system must recognize the difference between text and background within the image, filter noise out of the text information found and optionally calculate the skew angle of the text extracted.

Figure 11: An Example of Text Hidden in a Picture

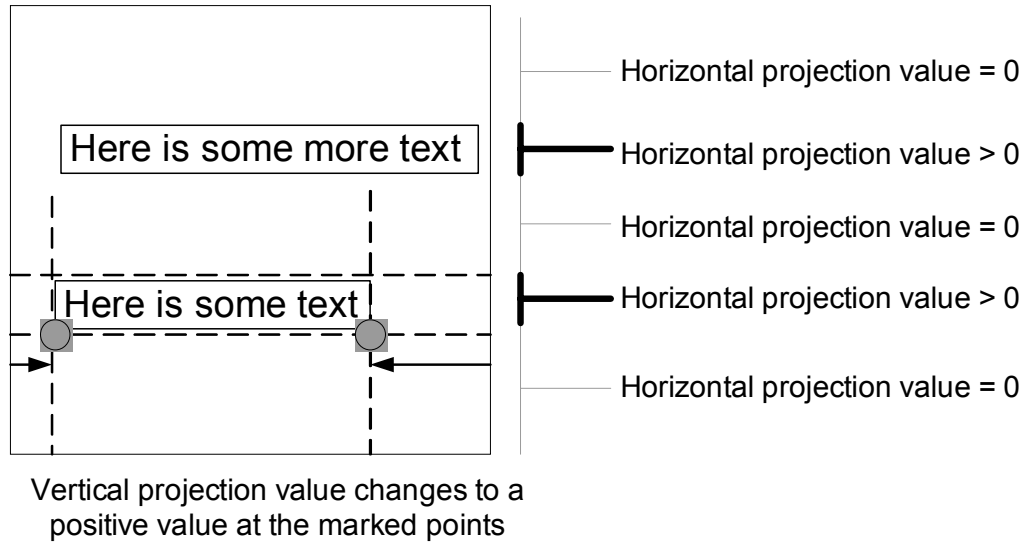


In image files such as the one displayed in Figure 11 where text is embedded in or superposed on other objects, the task to find text regions is a challenging one. The fact that the background cannot be assumed to be the same colour or texture throughout the image further complicates the issue. Wu, Manmatha et al (1997) have built a text detection system based on their observation that text possesses certain frequency and orientation information. They have also assumed that “characters of the same text string (a word, or words in the same sentence on the same line) are of similar heights, orientation and spacing” (Wu, 1997, p. 4). Gllavata, Ewerth et al (2004) have devised a scheme that uses a wavelet transform of the image in order to do edge detection. This

process is used to differentiate text from other objects. Details about the methods mentioned above can be found in (Wu, 1997) and (Gillavata et al, 2004) respectively.

Many black and white documents containing only horizontally aligned text are common in today's business world. Finding text in digital representations of these documents is a specialized case of the problem discussed above. This case will be the focus of the remaining OCR discussion in this paper. When these documents are scanned, it is useful to transform the image into the YIQ colour space. The brightness value Y of a pixel can be applied to a predetermined threshold value to determine if it is set (part of the image information) or not (part of the image background). Assuming that the orientation of the document is correct and noise has been removed, text regions can be found by calculating appropriate projections of the digital image as described by Parker (1997). A horizontal projection of the image is the number of set pixels in each row. It can be used to calculate the horizontal bounds of each line of text. While scanning through the projection results in either ascending or descending order, a horizontal limit of a line of text occurs when a transition between a zero and non-zero projection value is found (as shown in Figure 12). Using a similar approach, vertical projections can be used to find the left and right most extents of each text line. For each line of text, the vertical projection (the sum of activated pixels for each column) is calculated using the horizontal upper and lower limits found in the previous step. The left extent is found by finding the transition between zero and non-zero projection values starting with the left hand side of the document and moving right across it. Similarly, the right extent is found by searching for the same transition starting from the right side of the document and moving left.

Figure 12: Parker's Method Outline For Locating Text on White Paper



The probability of successful character recognition is dependent on how close the bitmap being considered is to the bitmap that it is supposed to represent. During the scanning process it is not always possible to get a perfect duplicate of the original document. Faulty or dirty hardware can introduce undesired information into the scanned document. Furthermore, the original document might have imperfections such as pen marks or dots, which can obscure characters found in it. These deviations from the original document are considered noises and character recognition systems are negatively affected by their presence. Parker (1997) proposes a few techniques to remove noise in scanned documents. He requires that multiple images of the document are obtained and assumes that the document is not moved between scans. If these conditions hold, noise can be reduced by assigning a pixel's brightness to be equal to the average brightness value for

that particular pixel across all samples. Alternatively, a voting process can be utilized where the sample pixels are deemed to be either set or unset utilizing a threshold value. In this case, the pixel is either assigned a specific value above the threshold if the majority of the samples are set or a value below the threshold otherwise. Parker (1997) also presents a median filter technique to address this problem. This method involves setting a pixel's brightness equal to the median brightness of the pixels in its defined neighbourhood. This approach only requires one scanned image but also reduces the contrast of edges. Unfortunately, this can also close gaps in characters or inadvertently join them together. In Parker's work (1997), he describes a kFill filter technique used to remove noise from scanned images.

Most methods used to classify glyphs rely on the image's text to be almost perfectly aligned horizontally. If the document is not fed into the scanning device at close to the correct orientation or if text inserted into the document is skewed, the preprocessing stage must reorient the text information so that it can be put through subsequent stages. Baird proposes a skew detection technique which is described in Parker's work (1997). It works on the premise that the bottoms of most characters are colinear. It specifies the following steps:

- Identify connected regions and assume they represent a character
- Find the bounding box for each region and locate points that identify the bottom edge of each box
- Given an angle θ , compute the horizontal projection of the points found above

- Maximize the function $A(\theta) = \sum_n P_i^2(\theta)$ where n is the number of bins

Figure 13: Baird's Method For Skew Detection In Text

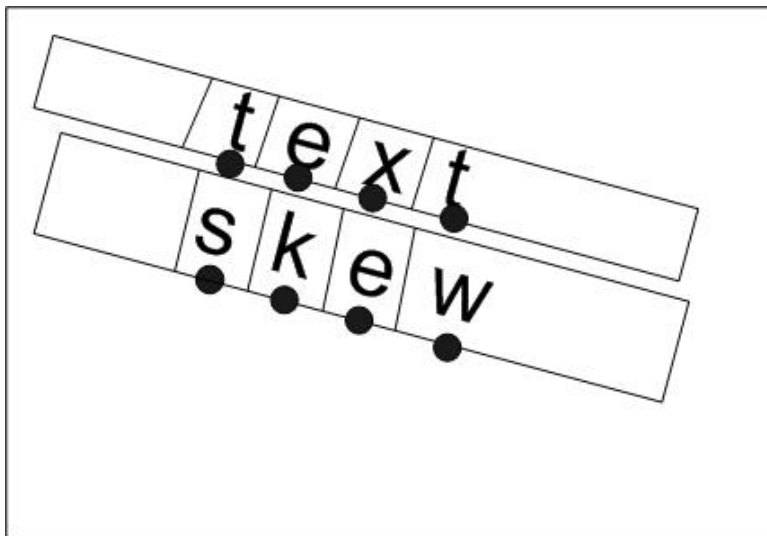
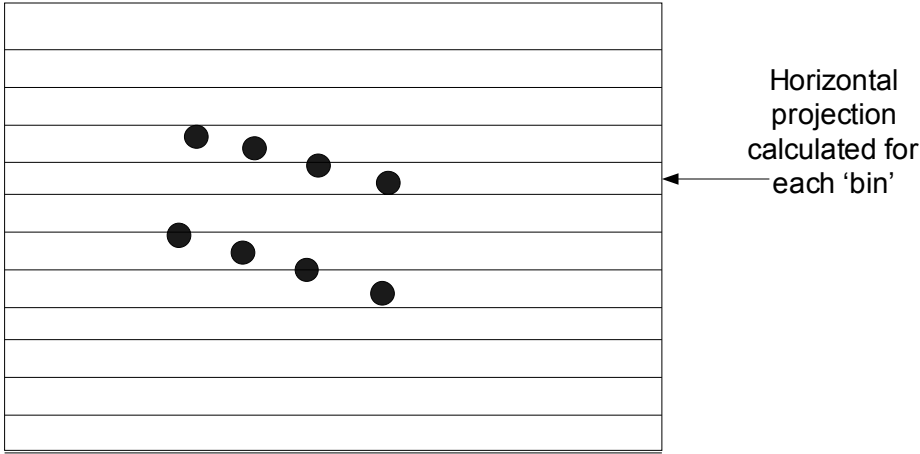
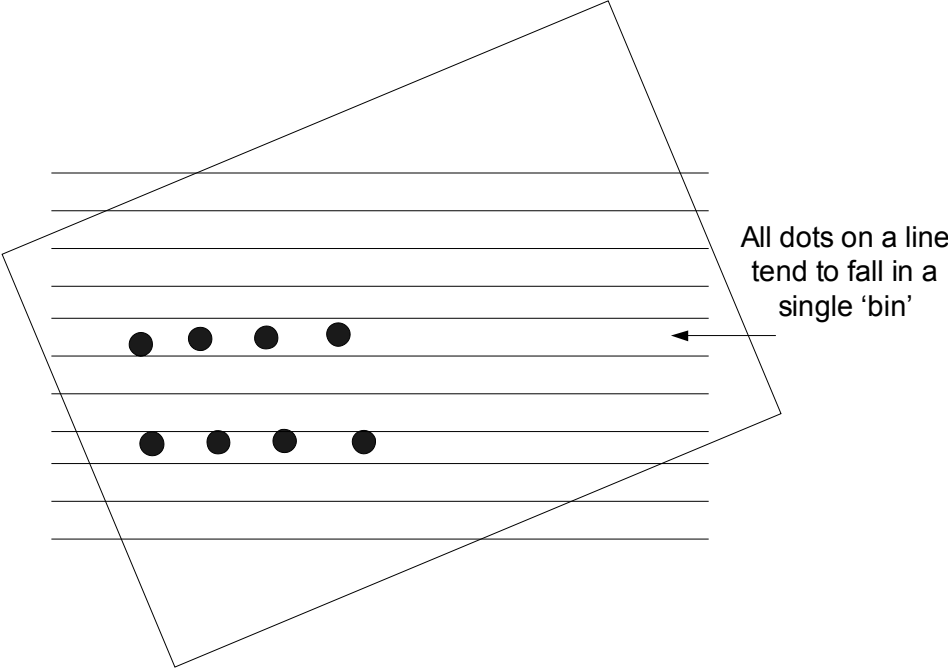


Figure 14: Using Horizontal Projections of Text for Skew Correction



a) 0 degree rotation



b) approximately 30 degree rotation (optimal)

Hashizume (Parker, 1997) has created a method based on a similar premise. In this technique, pairs of neighbouring characters are joined to obtain a best estimate of the skew angle.

Alternatively, the Hough transform can be used for skew detection. After doing the transform, it is observed that lines intersecting at the same point represent points that are on the same line in the spatial space. It is probable that a point where many lines intersect in the transform is a horizontal line which is parallel to text in the image. The co-ordinates of the intersection point can be used to calculate the skew angle.

After noise has been removed from the text regions and skew angle adjustments have been made, dividing the text region into individual glyphs is needed before character recognition can begin. Tsujimoto (Parker, 1997) attempts to approach the problem by calculating a break cost for each column. The break cost for a column is equal to the number of set pixels that have a set pixel in the previous column and the same row. Columns with low costs are good candidates for a vertical split. This works in many cases but is not perfect. “There is no algorithm that works in all of the cases that might be encountered while scanning any document of a reasonably large size” (Parker, 1997, p. 284). Context information can be used in addition to standalone algorithms in order to obtain reasonable accuracy.

Given that the individual glyphs have been extracted from the image at the preprocessing stage, each one can be analysed further to retrieve its feature information. Gatos,

Papamarkoes et al (1997) state that features should have the following properties: discrimination (different characters should have significantly different values), reliability (characters in the same class should have similar values) and independence (features should not be correlated with each other). Current recognition techniques fall into one of two categories: statistical or structural. These are further discussed in Appendix B.

In this section, an introduction to optical character recognition has been presented. It has outlined a general framework for how these systems are built and documented some basic techniques developed by researchers used to solve OCR problems. There are many advanced techniques beyond the scope of this paper that attempt to address difficult problems in the OCR field. Particularly challenging are the problems of recognizing handwritten characters and isolating text when the background has a non-uniform texture.

Security

Many computer applications require the exchange and storage of secret information. The level of secrecy depends on the value of the information to nefarious individuals.

Monetary value and individual privacy regulations are some reasons why parties do not wish to expose exchanged information publicly. The goals of security mechanisms might vary between applications, however one or more of the following attributes are desirable when transmitting transaction data over public networks (e.g. Internet).

- **Confidentiality.** In some applications, it is imperative that only authorized users can view transaction details. Ideally, data should be transmitted on a network that restricts access to valid users. On public networks, one cannot guarantee that confidential data is not being intercepted by unauthorized parties. As an acceptable compromise, however, data can be scrambled in a way such that it is statistically unlikely that someone other than the intended recipients can unscramble it correctly.
- **Integrity.** While transaction data is being transferred, it should be impossible for details of the transaction to be changed without the modification being detected.
- **Non-repudiation.** In many systems, a transaction represents a binding legal agreement between two parties. Such agreements require some type of explicit or implied confirmation. Any communication method used to transfer transaction data should guarantee that if a party has generated a confirmation, it cannot be

revoked or questioned at a later point in time. For example, assume a stock trading system user named Alice requests that a particular stock of hers be sold. Once she confirms her intentions and communicates them to her broker, the system should ensure that Alice cannot later claim that she never asked for the stock sale. In essence, it is imperative that the transaction confirmation is permanently bound to the original transaction.

Digital Watermarking

Digital watermarking is the act of modifying an electronic file in a manner such that its authenticity can be verified at a later point in time. It is used to add security attributes to a digital file. An electronic watermark has been defined as “an imprint in a document file that you can use to prove authenticity and to minimize the chance of someone counterfeiting the file” (Cole 2003, p.69). Digital watermarking is closely related, but not equivalent, to steganography; the act of concealing information within a message. In this section, digital watermarking will be discussed in detail with an emphasis on currently used methods and attacks used to defeat them.

Attributes of Watermarking Schemes

Watermarking techniques can be classified based on properties of the watermark and the file it is embedded in. Throughout the rest of this section, the file being protected by the watermark will be referred to as the cover work. Cover works typically include, but are not limited to, digital images, video footage and audio files. A marked cover work will refer to a digital file that has a watermark embedded in it.

The first classification discussed is the distinction between robust and fragile watermarks. Robust watermarks are designed such that they cannot be extracted from the cover work without severely damaging it. This type of watermark is useful for verifying ownership. If an owner can place a robust watermark identifying ownership in a cover work upon creation, theoretically no one else can later claim that they were the work's rightful owner. Any useable copy of the cover work will contain the original owner's watermark. Furthermore, in applications where the cover work's reading and copying devices can be controlled, robust watermarks can enforce copy controls. In such a model, all devices within the system are equipped to recognize watermarks of a defined format and will only process files with valid watermarks within them. Fragile watermarks on the other hand are used to detect modifications to a cover work. The embedded watermark is constructed based on some set of characteristics from the cover work itself. Any cover work will generate a unique watermark. The construction is designed so that when the authenticity of the cover work is questioned, the watermark can be extracted and compared with its characteristics. If the cover work does not match the extracted watermark, one can deduce that it has been altered. Semi-fragile watermarks relax the uniqueness constraint to accommodate non-malicious file manipulation such as data compression or image cropping.

When watermarks are embedded into a cover work, they introduce noise and reduce the work's fidelity, or perceptual similarity to the original. This fact becomes especially important when the cover works are multimedia in nature. Certain applications require

watermarked works to have a high level of fidelity in order for them to be useful.

Watermarks can also be described as being either visible or invisible. Invisible watermarks cannot be detected by the human perception system, but can be detected by machines. These marks are steganographic in nature because they conceal the presence of the watermark altogether. They tend to have a high level of fidelity because there is only a limited amount of information in the cover work that can be altered without being noticeable by the human eye. Conversely, it is difficult to make invisible watermarks robust, as robust watermarking techniques tend to require significant modifications to the cover work. This type of change is more suitable for visible watermarks, where the watermark is easily detected by human vision. Visible watermarks can also act as a deterrent for attackers, since they are aware of the fact that the cover work is protected.

Informed, or non-blind, watermarking systems use the original cover work at the watermark extraction stage. “This often substantially improves detector performance, in that the original can be subtracted from the watermarked copy to obtain the watermark pattern alone.” (Cox, Miller et al, 2002, p. 29). In applications where a goal is to verify ownership, the original work without the watermark is only available to the owner; otherwise an attacker can potentially watermark the original copy and claim to be its rightful owner. In these systems, the watermark detectors must not require the original, unmarked work in order to extract a cover work’s watermark. These systems are commonly referred to as blind detection systems.

Watermarking Defined

Pan, Huang et al. (2004) explain the watermarking process in mathematical terms. If the embedding process is a function ε that maps an unmarked cover work, x , and a watermark, w , to a marked work, x' , the process can be expressed by the equation in Figure 15. The optional variable, k , represents a key used if the watermark information is encrypted.

Figure 15: Mathematical Representation Of The Watermarking Embedding Process

$$x' = \varepsilon(x, w, [k])$$

The watermark detection function, δ , is modeled using the equation in Figure 16. The system's goal is to determine if some work x'' contains a watermark or not. The original unmarked work x is required for non-blind detection systems. The key, k , is a decryption key if the watermark searched for is in an encrypted form.

Figure 16: Mathematical Representation Of The Watermarking Detection Process

$$\{yes, no\} = \delta(x'', [x], [k])$$

This will be a decider but will not result in the watermark being extracted. An extraction function, β , will result in watermark w' being extracted from some work, x'' . As in the previous examples, x and k are optional depending on the type of watermarking technique used.

Figure 17: Mathematical Representation Of The Watermark Extraction Process

$$w' = \beta(x'', [x], [k])$$

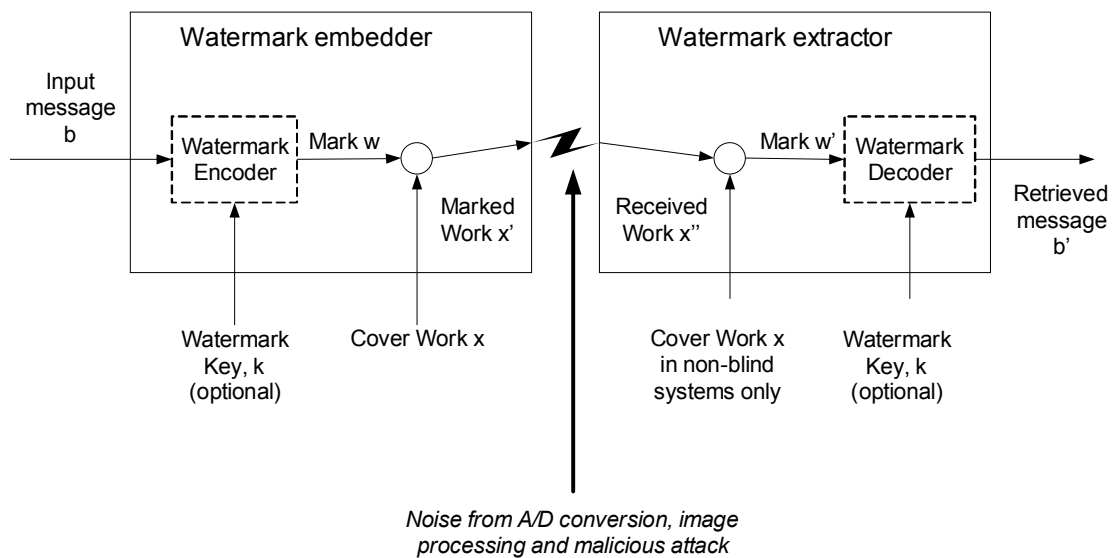
Assuming that a watermarking system's embedding function (modelled in Figure 15) is used to obtain a marked work x' , running its associated extraction function (in Figure 17) should yield the original watermark w (i.e. $w'=w$). In practice, however, this is not always possible because the marked cover work x'' used in the decider and extracting functions is often an approximation of x' . Figure 18 is an illustration of a generic watermarking system described above. Noise is often introduced into x'' when approximating x' due to imperfect analog to digital conversion operations, common image processing operations and/or intentional attacks designed to defeat the watermark. Differences between x' and x'' are referred to as distortions. Common distortions of image files are either valumetric or geometric in nature. Valumetric distortions affect the attributes of individual pixels. The following are examples of valumetric distortions to a cover work (Cox, Miller, Bloom, 2002).

- addition of some noise signal
- changes in brightness or contrast
- linear filtering (blurring/sharpening)
- lossy compression
- quantization

Geometric distortions refer to pixels being placed in incorrect positions when generating x'' . Rotation, scaling and translation operations done while approximating x' are common causes of this type of distortion. Often printing and scanning operations unintentionally affect cover works in this fashion.

Information coding and decoding are optional steps added into the embedding and extraction functions respectively. This gives users the flexibility of representing watermark information in an arbitrary manner.

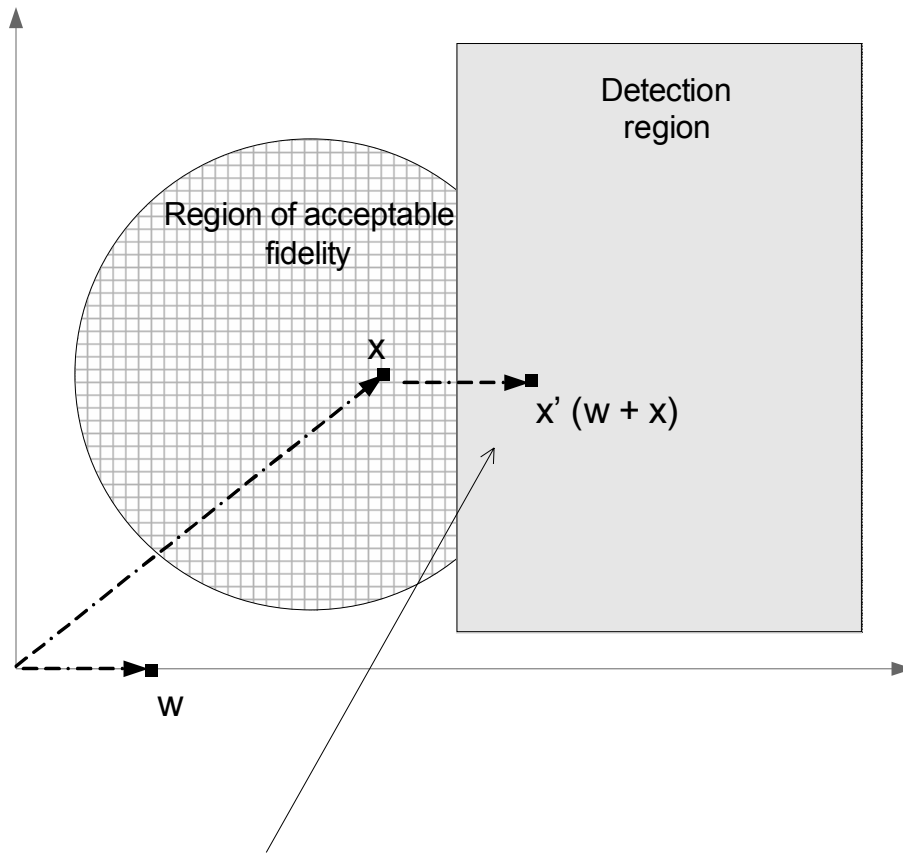
Figure 18: A Generic Watermarking System Design



Cox, Miller and Bloom (2002) present a geometric view of watermarking systems. In this model, every possible work is mapped to a unique vector in an n -dimensional media space. In digital images, n is equal to the number of pixels stored times the number of attributes stores per pixel. For example, with colour images stored in RGB format, $n = 3$

* number of pixels. The watermarking system in turn divides the space into specific regions of interest given a particular work. The region of acceptable fidelity defines a set of vectors which represent marked cover works that are humanly indistinguishable from the original. “It is extremely difficult to identify the true region of acceptable fidelity around a given Work, because too little is known about human perception.” (Cox, Miller et al, 2002, p. 62). Researchers, however, have devised ways to estimate this region. Given a watermark w and a key k , the detection region is the set of vectors that would successfully yield w when processed by the watermarking system’s extraction algorithm. The intersection of regions mentioned defines a set of works deemed to be successfully watermarked by the system. Figure 19 summarizes these relationships. The goal of the embedding function is to find some vector v that when added to the original cover work’s vector o , creates a vector c that represents a valid marked cover work containing w .

Figure 19: A Geometric View of the Watermark Image Space



Region where successfully marked cover works lie

Earlier in this section, it was mentioned that information contained in some watermark w is optionally put through some coding process before it is embedded into the original work. While it is true that a binary message can be placed directly into a work, a coding method is often utilized to improve a cover work's robustness, fidelity or code separation properties.

Waveform-based detectable watermarking techniques map possible codes to waveforms. Given a set of codes B , where $B = \{b_1, b_2, \dots, b_n\}$, some coding rule Φ maps each $b \in B$

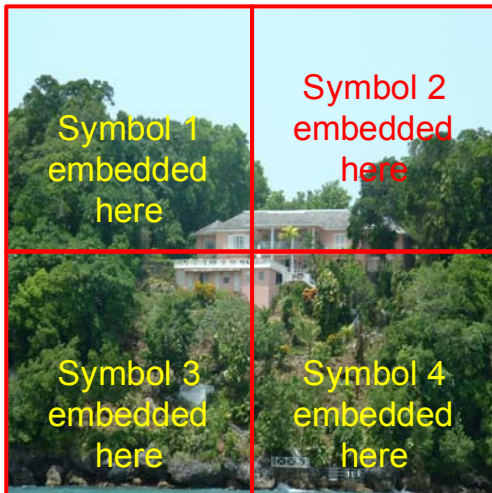
to a distinct member of some waveform set W , where $W = \{w_1, w_2, \dots, w_n\}$. Researchers have devoted significant efforts into defining an optimal set W and function Φ . Barni and Bartolini (2004) outline a method for defining W where each value w_x , is a pseudo-randomly generated sequence drawn from a probability density function.

In a direct embedding system, each possible message m is mapped to a unique code b_n . Embedding m into the original work is now simply a matter of adding the work's vector and b_n 's associated watermark w_n . Watermark detection is a matter of comparing the work to each of the n watermark vectors in W . This comparison produces a value estimating the probability that mark w has been embedded into the work being examined. The w_n (and thus the message m_n mapped to it) that yields the highest detection probability is deemed to be embedded in the work once this probability falls within a pre-defined threshold. If all probability values fall below this threshold, it is deemed that no watermark is in the work.

In informed embedding approaches, the embedding application uses information about the cover work when devising the mark actually inserted. This allows the system to create a mark is most suitable for the application in question. Often, obtaining strong watermark robustness and ensuring that marked works retain perceptual similarity to the original are contradictory goals. Informed embedding applications can have multiple watermarks w_n associated with a particular code b_a . By utilizing the cover work's properties, the embedding algorithm can choose the w_n that best suits its goals.

Assigning a unique mark to each message is simple however this technique is impractical when there are a large number of possible messages. Embedding a small piece of information such as a postal code or driver's license number would require generating thousands of unique watermarks. Large numbers of messages can be represented by defining each code b_n to be a symbol in some alphabet \mathbf{A} . Let $|\mathbf{A}|$ be equal to the number of symbols that are contained in \mathbf{A} . If the system is defined to support messages which are represented by sequences of up to x codes, then $|\mathbf{A}|^x$ messages can be embedded in a work. For example, this allows a system which supports 4 unique symbols and ten character sequences to represent 1048576 different messages. Each code b_n is assigned a unique watermark w_n as described earlier. A set of watermarks representing a message m is defined to be $\{m_1, m_2, \dots, m_x\}$, where $m_a \in W$ and $0 < a < (x+1)$. In order to perform the embedding process, the system first divides the original work into n disjoint regions. This can be done in either the spatial or frequency domain. Each region is designated a symbol position or index value between one and x . After this process is complete, each symbol m_a is embedded into the region designed to index a . Figure 20 explains this process. In the previously mentioned direct embedding case, each unique watermark vector would have to be compared with the work being examined in order to determine which watermark, if any, has been inserted into it. By using a sequence of symbols to embed the message as described above, only $|\mathbf{A}| * n$ comparisons are needed.

Figure 20: Division of the Image Space Into Disjointed Blocks For Watermarking Purposes



Watermarking Techniques

Watermarking techniques differ by the domain in which they embed marks into cover works. Domains are carefully chosen because they ultimately determine attributes of the system such as robustness and computational complexity.

Cover works can be marked in the spatial domain. Since most images are natively represented in this domain, spatial watermarks tend to be computed very quickly. Marks are embedded into a cover work by slightly modifying the luminance or colour of a known set of pixels in order to satisfy defined encoding rules. Detectors extract the mark by observing the set of pixels in question.

Katzenbeisser and Petitcolas (2000) describe a popular marking technique called Least Significant Bit substitution, or LSB, in their work. Assuming that the mark can be represented as a binary sequence of length n , the system selects an ordered set of n pixels to evaluate. Both embedding and detector processes must be aware of or be able to derive the selected pixel set. This can be done in some arbitrary manner (i.e. consecutively or pseudo randomly). In order to embed a mark using this technique, the embedding system modifies the pixels as follows. The binary representation of luminance for the n^{th} pixel is modified such that its LSB matches the n^{th} bit of the message. To extract the mark, the detector simply retrieves the LSB of pixels' luminance values defined in the predetermined pixel set. Since the luminance values are not significantly changed, the marked work's fidelity remains relatively high. The capacity of such a scheme is large as well since one bit of information can be stored for every pixel in the original work.

In another related spatial watermarking technique, a work is divided into n ordered, disjoint subsets as shown in Figure 20. The embedding is done by ensuring that the parity of the luminance LSBs for all the pixels in the n^{th} set is equal to the n^{th} bit of the message. Katzenbeisser's and Petitcolas's (2000) work presents other spatial techniques based on quantization and dithering.

In general, spatial techniques are only appropriate for fragile watermarks. Typical operations such as compression or translation can completely destroy a spatial based mark. If a watermarking system's goal is to provide robust watermarks, it must generate marks that are invariant to most common types of distortions.

Frequency domain techniques have been the subject of most current research. The Discrete Fourier Transform (DFT) mentioned earlier in this paper provides the basis for approaches of this nature. Unfortunately, DFT becomes problematic because its calculation does not always result in real numbers. Usually, derived forms of the DFT such as the Discrete Cosine Transform (DCT) are used in practical watermarking systems (Katzenbeisser & Petitcolas, 2000). Marks embedded in the frequency domain tend to survive common compression methods such as JPEG and MPEG. Since DFT magnitudes are not affected by translation, watermarking techniques based on DFT algorithms also tend to be immune to translation attacks. Since wavelets add the concept of spatial location to frequency information, Discrete Wavelet Transform (DWT) systems add the additional flexibility of embedding marks in favourable positions within a work.

Embedding a mark in the frequency domain is done by modifying the magnitude of various frequency coefficients in the cover work (Nikolaidis & Pitas, 1999). Ó Ruanaidh and Pan (1998) present an example of this based on the DCT. In other frequency domain-based systems, such as the ring-shaped watermark approach (Nikolaidis & Pitas, 1999), this is accomplished by creating defined relationships between certain coefficients after calculating the frequency domain representation of an image.

More complicated systems address the more general problem of creating watermarks invariant to all geometric attacks. The equation below is an affine transformation of the

point (x_0, y_0) to (x, y) . It encapsulates the geometric operations of translation (via the $[e \ f]$ matrix), scaling (via the $[a \ b \ c \ d]$ matrix) and rotation (via the $[a \ b \ c \ d]$ matrix).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

One approach to solving this problem is to reverse any distortions performed on the marked work before it is presented to the extraction mechanism. Various permutations of the affine transformation parameters can be used to derive the undistorted work. Two problems exist with this approach; complexity and the possibility of getting false positives (Katzenbeisser & Petitcolas, 1999). Given the search space of six independent variables, calculating every possible transform and running the result through an extraction algorithm is computationally expensive. In addition, the larger the search space, the more likely a transform will yield a valid but incorrect watermark. If the marked cover work is known to have reference points, this fact can be used to significantly reduce the size of the search space.

An alternative approach is to ensure that the watermarking system is invariant to all possible affine transformations. As stated earlier, DFT schemes tend to be invariant to translation, however, their watermarks can be destroyed by other geometric operations. Watermarks embedded using the Mellin-Fourier Transform have been noted to be invariant to affine transformations. This process involves a DFT transformation followed by a log-polar mapping. The results of this operation are subsequently put through a second DFT process to arrive in the invariant marking space. Ó Ruanaidh and Pun (1997)

provide more detail about the Mellin-Fourier Transform in their works. This appears to be an optimal solution however watermarks embedded in this space are not resistant to lossy image compression (Ó Ruanaidh & Pun, 1997). In addition, the log-polar mapping and its inverse performed during the Mellin-Fourier Transform degrade the fidelity of the work.

Xin, Liao et al (2004) present a geometric-invariant watermarking scheme based on Zernike moments. Their approach embeds watermarks by quantizing the magnitudes of Zernike moments within cover works. They conclude however that significant improvements must be made to this approach for it to be practical. Chen, Yang et al's (2005) work noted high computational costs associated with calculating high order Zernike moments which could also hinder its use in real world watermarking systems.

Increasing the Robustness of Watermarks

As shown in Figure 18, a marked cover work can be slightly altered during transmission to a watermark extraction device. When the watermark is obtained, differentiating between unavoidable noises such as rounding or conversion errors and modifications added by an attacker is desirable. This problem can be addressed by introducing redundancy into the codes that are being transmitted. The result is that not all possible codes are valid. Furthermore, valid codes are selected so that they are “far” from each other. If minor errors are found, they are expected to become invalid codes. These invalid codes can then be mapped back to the “closest”, valid one. For example, Hamming codes are “separated” by three bits and can successfully detect single bit errors.

Trellis and turbo codes allow detection of more significant errors. Cox, Miller et al's book (2002) provides an excellent example of how Trellis codes can be implemented.

Another way of improving robustness is to place duplicate watermarks within the same work. The extraction process could retrieve all the watermarks and compare them. Any changes could be corrected using a normalized version of the watermarks. Alternatively, some voting scheme could be used to analyse the watermarks on a per-pixel basis. The system would try to determine what each pixel of the original watermark is based on viewing the corresponding pixels in each of the extracted marks. Decisions could be made on a "majority rules basis" or possibly some other threshold based on the robustness required.

If a watermark stores its message in an image format as oppose to a binary one, significant robustness improvements can result. The watermark extraction process can utilize OCR techniques discussed earlier in this paper to disregard distortions added to marked cover works.

All of these techniques add robustness at the cost of channel space. Cover works have a finite amount of space they can use to transmit watermarks. Redundant data added to achieve robustness uses up room that could be used for longer messages.

For watermarks to be robust, they “must be embedded in the perceptually significant components of the image.” (Ó Ruanaidh and Pun, 1997, p. 536). This is particularly true in the frequency domain.

“...most forms of image processing damage the high frequencies in an image’s Fourier representation. That is, lossy compression quantizes many high frequencies to zero, halftoning adds high-frequency noise, scanning applies a low pass filter before sampling and so on. It is almost certain, therefore, that any watermark information placed in high frequencies will be damaged in a processed image” (Cox , Miller et al, 2002, p. 245)

Low frequencies are also problematic location to embed secret marks. They are easy to detect there. Altering a work in this region might degrade the fidelity of the work to the point where it is unusable.

Intuitively, most compression techniques remove data from areas that humans will not notice. Determining these areas is an outstanding problem and relies on an understanding of the human visual system. Nikolaidis and Pitas (1999) discuss this topic further in their work.

Watermark Security

Kerckhoff’s principle asserts that the secrecy of a watermark should be dependant on the protection of the private key used during its generation. Furthermore, the algorithm used to create the watermark must be able to be made public without compromising the watermark’s security.

Many current watermarking algorithms divide an image into ordered, disjoint sets of pixels. Typically, a binary bit of information is embedded into each selected set.

Pseudorandom number generators can be used to select the sets which store generated watermarks. If the watermark embedding and detection processes are the only entities aware of the number generator's seed, then attackers will have a difficult time figuring out where watermarks are being stored in images of interest.

Common encryption techniques can also be used to secure a watermark. Even if an attacker successfully extracts a watermark from a marked cover work, its utility is diminished if it cannot be decrypted. A brief introduction to cryptography is provided in Appendix C.

Security of a watermark is important when it is used to confirm a transaction or to identify ownership. Security schemes for watermarks are designed to thwart some of the attacks mentioned in the following section.

Watermarking Attacks

Attackers can use noise to distort a cover work for malicious purposes. One reason to attack a cover work is to defeat read or copy protection schemes. In multimedia copy protection applications, a watermark is inserted into the cover work to define how playback devices can access it. An attacker can circumvent this protection by removing this watermark from the cover work altogether. Another reason to attack a work is to illegally claim ownership of it. A malicious user could attempt this by inserting a

watermark identifying himself or herself as the rightful owner of the cover work in question. Even if the original owner's mark is not removed, both parties can legitimately claim ownership of that work. In the following paragraphs, methods of attacking watermarks discussed in Cox, Miller et al (2002) will be described.

Scrambling attacks are used to elude watermark detection. In this attack, the marked cover work is divided into small pieces and scrambled before presentation to the watermark detector. Assuming that the pieces were scrambled in a deterministic fashion, they can be put back together after the watermark detection stage.

Attacks can be made to a marked cover work via image processing operations such as shearing, cropping, filtering or noise removal processing. In these types of attacks, the detection tool processes the work and any embedded watermark within it goes unnoticed.

In a copy attack, one tries to copy a watermark from one cover work and places it in another. This type of attack is used to convince a watermark detector that the second work was marked rightfully. First, the attacker uses a watermark removal technique on the marked cover work, m_1 , to approximate the original unmarked work o_1 . Define this approximation of o_1 to be o'_1 . The difference between m_1 and o'_1 yields an approximation of the work's watermark w_1 , defined as w'_1 ($w'_1 = m_1 - o'_1$). The watermark w'_1 can be inserted into some unmarked work o_2 by adding the two vectors ($c_2 = o_2 + w'_1$). If o'_1 is a "good" estimate of o_1 , then the extraction or detection system processing the newly marked work c_2 will believe that watermark w_1 is embedded in it.

When watermarks are added to cover works in order to verify ownership, ambiguity attacks can be utilized to weaken the strength of such an assertion. In this attack, a malicious user generates a watermark that is likely to already exist in a marked cover work. If this is done successfully, both the rightful party and the attacker have equally valid claims of the cover work's ownership.

More advanced techniques such as gradient descent and sensitivity analysis attacks assume that the attackers know the detection region in the media space. This is used to calculate a way to minimally alter the cover work such that the embedded watermark can no longer be detected.

Digital watermarking can be a robust method of securing digital files. The goals of a watermarking system can range from guaranteeing file ownership to verifying file integrity. Unfortunately, the goal of robustness must be balanced with an end-user's desire to have the marked work be as perceptually similar to the original as possible.

CHAPTER III

METHODOLOGY

Proposed Solution Design

By storing exam documents in digital format, exchange between the parties in the exam system discussed can be accomplished using the Internet. The following is a description of an exam exchange system that would run with documents in digital format.

An exam created by a professor can be converted into a series of digital files. It is assumed that the professor created the file using some word processing software capable of having its files converted into image files. A watermark inserted by either the University, its faculty or its proctoring institutions will contain the attributes noted in Table 1. If the professor is submitting the exam from a remote location, it can be sent to the University via a Secure Socket Layer (SSL) communication channel to ensure that the exam contents remain confidential. SSL communication channels are used for any file exchange between remote parties within the system. Upon receipt, the University stores the exam until it is sent to the proctoring sites for student exam sittings.

Table 1: Fields for Exam Watermark

Hash Value	Calculated based on the ASCII values of pre-printed text in the exam document
Integer Pairs	Represents the location of pre-printed text mentioned above in terms of percentages. The first value is determined by computing the ratio between the distance from the top edge of the document to the upper extent of the text and the length of the document.

	Similarly, the second value is computed by calculating the ratio between the distance from the lower extent of the text to the top of the document and the length of the document.
Examination ID	A unique identifier for the exam document. It can be a combination of attributes such as the course number, the sender id or the date exam is to be administered.

When an exam is sent from the University to a proctoring site, it is put through a watermark extraction system upon receipt. The watermark is inspected to ensure that the exam is indeed sent from the University. Furthermore, the watermark's contents are used to verify the authenticity of the exam document. Assuming the watermark is legitimate, the exam is printed for student use during the appropriate exam sitting. A student handwrites responses to the exam questions on the printed copy during an exam sitting and returns the paper to the proctoring institution. Before returning the exam to the University for marking, the paper files are transformed into digital format using a scanning device. Assuming that the watermark is not destroyed during the transfer or the student writing process, the validity of the document can be rechecked. A new watermark is inserted into the file which adds the proctor's identity information. Upon arrival at the university, a similar verification process takes place. The credentials in the watermark are scrutinized by a receiving application before storing the exam in a persistent data store. This process is summarized in Table 2.

Table 2: Outline Of The Proposed Exam Watermarking System

Processing Stage	Processing Description
Embedding Step	<ol style="list-style-type: none"> 1. Generate a hash value using the Examination ID specified 2. Print the Examination ID in a predetermined location on the exam

	<ol style="list-style-type: none"> 3. Encrypt the hash using the Extractor's public key, 4. Create an image of the encrypted hash value 5. Embed the watermark image into the exam document using a chosen marking algorithm.
Extractor Step	<ol style="list-style-type: none"> 1. Retrieve watermark from the incoming exam document. 2. Run watermark through an OCR engine to obtain its ASCII representation. 3. Decrypt the all embedded information within the watermark using the Extractor's private key 4. Retrieve Examination ID from predetermined location on the exam image. Use OCR engine to obtain its ASCII representation 5. Use the ASCII representation as input to the hash algorithm used. 6. Compare the hash result, with the hash value originally sent with the document (Step 3). If they are equal, the information has not been tampered with.

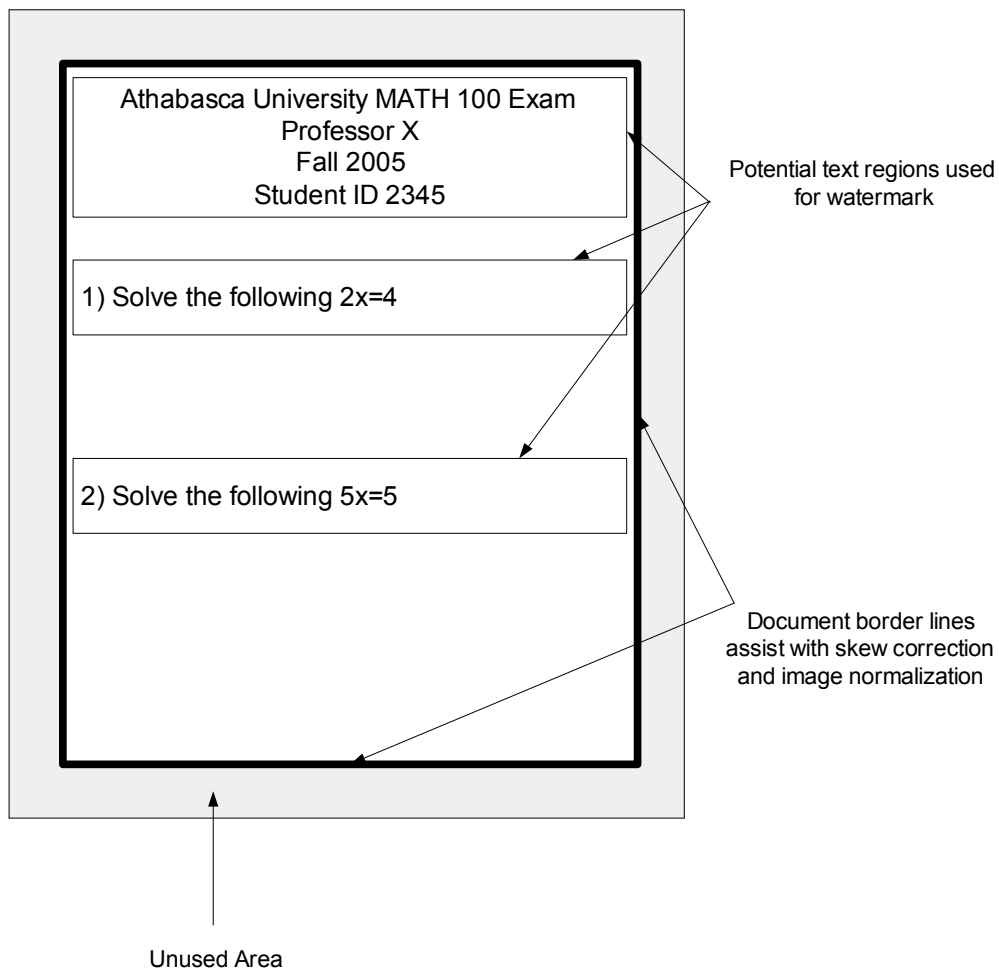
When a grader downloads an exam, prints it and adds markings for grading purposes, the documents must be rescanned to re-enter the system. The same set of steps listed for the proctoring institution above must be redone. The only difference is the contents of the watermarks in the process. In this model, multiple watermarks can be present in the document at the same time. Each watermark within the document must be stored in an independent area of the document so that the watermarks do not overwrite each other. For additional security, when an exam is stored in a persistent store it should be both password protected and encrypted.

Exam Document

Each page in an exam document will be generated using a predefined template. This will allow the system to perform skew correction and normalization when converting exams

into electronic images from physical paper. Figure 21 shows a picture of the structure put in place.

Figure 21: A Sample Exam Document



The document's usable area will be defined as a fixed percentage of the actual image size. During the scanning process, capture errors can result in the actual document being off centre or slightly altered in size. Minor errors of this nature can be compensated for by using scaling and translation image processing techniques without significant loss of data.

Similarly, the document border lines which separate the usable and unusable areas of the document provide a convenient means of correcting skew errors caused by not perfectly aligning the physical document with the scanning device. The horizontal border lines are guaranteed to produce the maximum possible horizontal projection for this document. Knowledge of this fact can be used to correct minor rotation errors made during image capture. Skewing algorithms mentioned in the literary review section are possible solutions to this issue.

Text found in the exam document is used to formulate the watermark as mentioned earlier in the solution design. This is done so that blind watermarking techniques can be used within the system. Blind techniques are preferable in this solution as the University will be receiving exams from multiple proctoring institutions. Limiting the need to expose data to the proctor sites makes the system more secure.

Watermark Experiment

Overview

A significant barrier to making this solution a commercially viable one is the possibility that digital watermarks will not survive the temporary transfer from digital format to physical paper. While an exam document is on paper it is subject to being marked, soiled or torn. A system of this nature cannot be overly sensitive to such modifications. Moreover, office quality scanning and printing devices are not guaranteed to perfectly preserve all the information found in the original exam document. The system described

in the previous section attempts to address this by interpreting the watermarks using OCR technology. Noise reduction processing and feature based matching done during the OCR process can reduce a watermark's sensitivity to errors.

Objective

To determine if digital watermarks designed in image files can reliably survive the transformation from digital representation to physical form. If the results show that this is possible, a secondary objective is to determine if such a solution is commercially viable.

Tools

- PC
 - AMD Athelon 1.8Ghz CPU
 - 512 MB RAM
 - Windows XP Professional operating system
- Printer
 - Konica Minota PagePro 1350W
- Paper
 - White, letter size paper (20 lbs)
- Marking Devices
 - Black pen
- Graphics Application
 - Microsoft Paint
- AU Watermarking Test Application

- DCT Based Watermarking System
- Asprise OCR Software or similar application (OCR software is not included with the included source code)

AU Watermarking Application

A watermarking application has been developed specifically for this research project. Its extraction and embedding implementations are based on a marking algorithm described in Katzenbeisser and Petitcolas's (2000) work.

When the application receives an embedding request it creates a copy of the cover work. This work is divided into 8 x 8 pixel blocks and an ordered set of these blocks is selected pseudo randomly to hold the desired watermark. To ensure that all pixels in the cover work are within a block, black pixels are added to the right side or to the bottom of the cover work as required. This ensures that the height and width of the document are an even multiple of 8. One bit of information is stored per block. Each block has its associated pixels converted into the frequency domain using the DCT. Two specific values within this domain are selected to store a binary piece of information. In particular, if the value stored at location (4, 1) is greater than the value at location (3, 2), then the watermark bit embedded is a one, otherwise it is a zero. The system enforces this relationship by swapping the two values in the frequency domain if the current values do not reflect the desired watermark bit values. Once all of the watermark bits are embedded

in the selected blocks, the modified blocks are converted back into the spatial domain and the complete image is saved to disk.

The actual watermark to be embedded into the work is dynamically generated by the AU Watermarking Application. It is composed of a bitmap representation of the specified message. The AU Watermarking Application reads the ASCII message passed into the system and then draws this message in the 12 pt. Arial font to a bitmap. The size of the bitmap in bytes is calculated and the result is added to the watermark before the bitmap's binary information. This is done because the AU Watermarking system is blind and requires the size information so that it knows how many blocks should be processed at the extraction stage.

Extracting the watermark is done by retrieving the same ordered set used in the embedding process. This is accomplished by using the same seed and pseudorandom number generator used at that time. Each block is converted into the frequency domain and the locations used to store the watermark information are checked to extract the embedded binary information. The first 32 bits of the watermark are reserved for size information. The extraction algorithm uses the first 32 blocks in the work to calculate the size of the rest of the watermark. At this point, the remaining blocks are processed to construct the bitmap image embedded in the work.

Procedure

1. Create test documents similar to the format in Figure 21. These should be stored in bitmap (.bmp) format.
2. Scan documents created in step 1) at the following settings: 200 dpi, 300 dpi, 600 dpi. Both greyscale and colour formats should be captured. Note the time it takes to scan each document and the resulting document file sizes.
3. Use the AU Watermarking Test Application to embed watermarks into files generated in step 2). Each document in 2) should result in 8 output files; 4 watermark output files and 4 marked versions of the original as described below:
 - a. Original document with a watermark containing the 4 character string “test” embedded.
 - b. Bitmap of the string “test”
 - c. Original document with a watermark containing the 10 character string “testtenchr” embedded
 - d. Bitmap of the string “testtenchr”
 - e. Original document with a watermark containing the 40 character string “1234567890123456789012345678901234567890” embedded
 - f. Bitmap of the string “1234567890123456789012345678901234567890”
 - g. Original document with a watermark string containing the 52 character string
“abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ” embedded

h. Bitmap of the string

“abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

”

4. Ensure that watermarks successfully embedded can be extracted reliably using the AU Watermarking Test Application. Ensure this process is successful by ensuring that the extracted output file is equivalent to the corresponding bitmap file created in Step 3).
5. Make copies of each successfully marked work and perform one of the following manipulations per copy:
 - a. Use a graphics application to mark a soft copy the document. Use the colour black for marking.
 - b. Convert a soft copy of the document into jpeg format and back into bitmap format.
 - c. Print document and rescan image without marking it. Use the original scanner settings.
 - d. Print document and rescan image after marking it with a black pen. Use the original scanner settings.
 - e. Print document and rescan image after folding into 4 sections. Unfold before scanning. Use the original scanner settings.
 - f. Print document and rescan image after adding a significant tear to the document. Use the original scanner settings.

6. Put each copy through the AU Watermarking Test Application to extract the embedded watermark. Take the resulting files and run them through the selected OCR application. Document which files actually yield the originally embedded string.

Data Analysis Techniques

Since the sample size of the experiment is so small, the results will be statistically insignificant. The goal will be to simply draw general conclusions based on observations made during the procedure. Observations will be formulated based on the following information:

- Reports from the OCR software which state whether embedded watermarks are successfully retrieved or not.
- Error messages from the AU Watermarking Test Application. This information will help determine why failures occurred during the processing of unsuccessful extraction attempts.
- Image size information for cover works and watermark images.
- Timing data gathered during the procedure and from the AU Watermarking Test Application output

CHAPTER IV

RESULTS

This section summarizes the data gathered while following the procedure previously outlined. The results are organized into two sections. The first section presents data which is used to verify the technical aspects of the experiment. This will help answer the question, “Is the solution presented technically feasible?”. The second section will present data relating to the commercial feasibility of the solution. This will help address the secondary objective of the experiment. If the solution is technically possible but not practical, it should not be recommended as a means of replacing the current Athabasca University examination system. Each section will now be presented in turn.

Technical Considerations

The data gathered during Step 4 of the procedure is used to ensure that the watermarking system is working correctly. Table 3 shows the results of the actions taken. The image of the word “test” could be retrieved regardless of the resolution settings. The longer strings could not be embedded into images scanned at 200 dpi. The number of bytes required to hold the watermark images is displayed in Figure 22. These images were created in the 12 pt. Arial font. Originally an 8 pt. font was selected in order to reduce the file size of the watermark images being created. A decision was made to move to a 12 pt. font because the Asprise OCR engine used could not recognize glyphs generated using the smaller font size.

Table 3: Results of Running Character Recognition On Extracted Watermarks

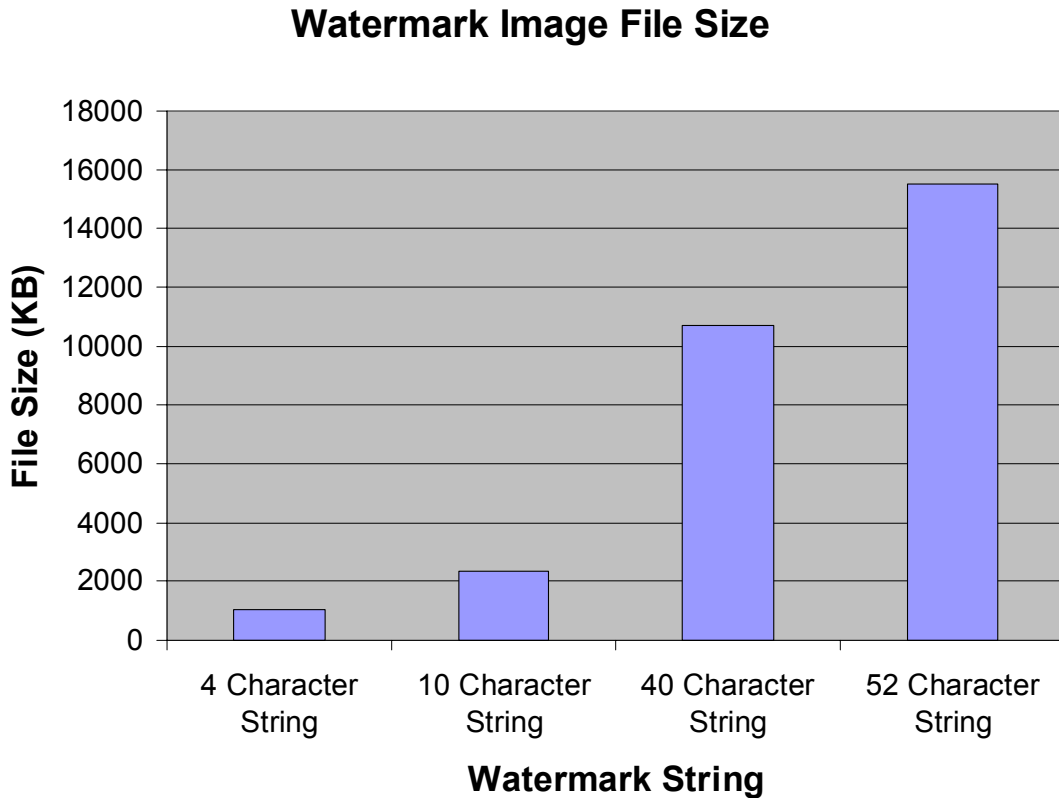
	4 Character String	10 Character String	40 Character String	52 Character String
200 dpi – Colour	√	?	x	x
200 dpi - Greyscale	√	?	x	x
300 dpi – Colour	√	?	?	?
300 dpi - Greyscale	√	?	?	?
600 dpi – Colour	√	?	?	?
600 dpi - Greyscale	√	?	?	?

√ - OCR read of extracted watermark was successful

? - OCR read of extracted watermark failed, but the watermark is humanly recognizable

x – watermark could not be embedded into the cover work because it is too large

Figure 22: File Size of Watermarks Created by the AU Watermarking System



Since the remaining steps of the procedure relied on having a watermark that is reliably extractable, Steps 5 and 6 were only performed on the cover works marked with the image “test”. The extraction system could not retrieve watermarks from any of the test images generated. The transition of the document to paper completely destroyed the embedded watermarks. The jpeg conversion process delivered similar results.

Failed tests usually caused the extraction process to throw a general protection fault.

Running the system in debug mode showed that the size portion of the watermark had its

most significant bits incorrectly read by the system. Given that the watermark reserves 32 bits for the bitmap's file size, a single bit error in the more significant bits can cause the file size value to be incorrect by millions of bytes. This type of error caused the system to attempt to request more memory than what was physically available to the system.

Tests run using images from Step 5 a) produced unsuccessful but less fatal results. Table 4 summarizes these results. A comparison of the extracted image with its associated watermark image obtained from Step 4 showed little damage to the watermark. However, the damage that did occur during the marking process rendered them unusable. Single bit errors to either the file size portion of the watermark or the BITMAPINFOHEADER structure within the bitmap image caused the extracted watermark file to violate the bitmap format rules. The high resolution documents seemed to be less damaged by the marking process.

Table 4: Results From Extracting Watermarks From Cover Works Marked By A Graphics Application

	File Size Value Correct?	Intact BMP header?	Valid BMP File Extracted?
200 dpi – Colour	x	x	x
200 dpi - Greyscale	x	x	x
300 dpi – Colour	√	x	x
300 dpi - Greyscale	x	x	x
600 dpi – Colour	√	x	x

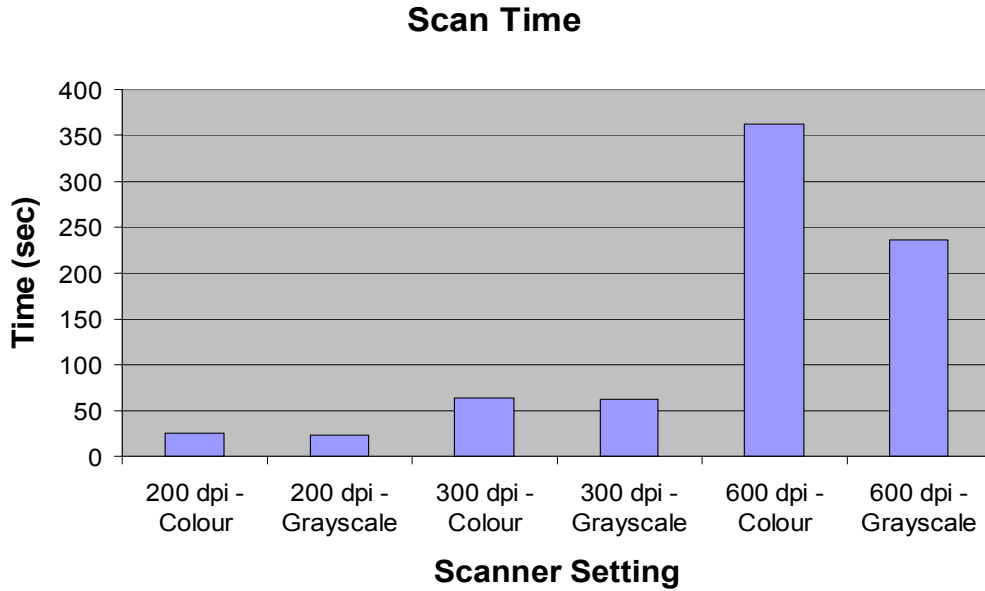


√ - positive answer
x - negative answer

Commercial Feasibility Considerations

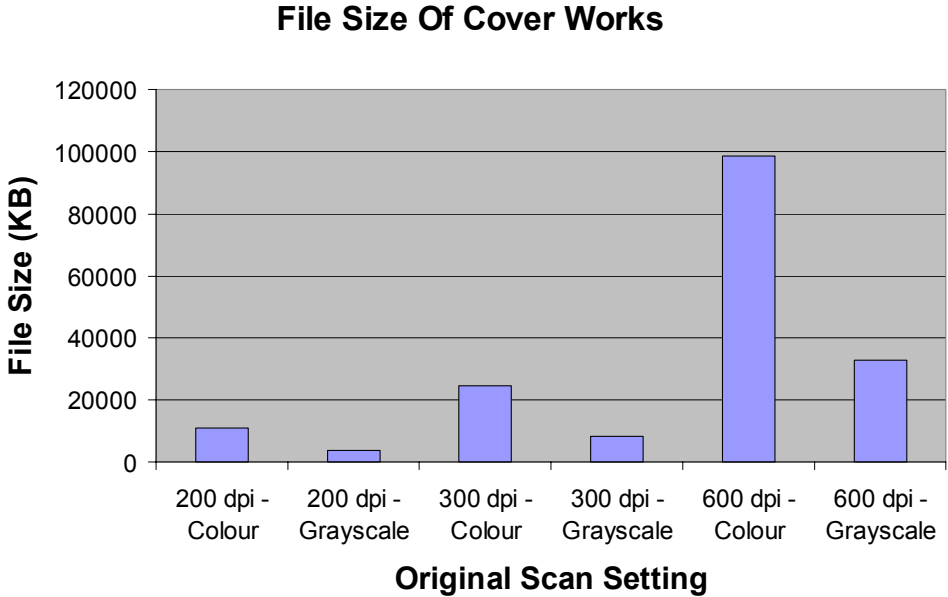
In order for this AU Watermarking System to be used as a basis for a production quality exam exchange system, it must not introduce an unreasonable amount of processing overhead. Each page of an exam must be scanned and a watermark must be embedded in the electronic exam exchange model proposed. The system used in the proposal took approximately 6.5 minutes to embed the 52 character string watermark into a single page. Figure 23 illustrates the time required for the scanning devices to scan a page. Given the fact that a 600 dpi image required about 2 minutes to extract the 52 character string watermark, a proctor would need at least 15 minutes to process each page of a student's exam. It is not uncommon to see exams that are more than 20 pages in length.

Figure 23: Time Required To Obtain a Letter Size Image From The Scanner



Once the cover work images are obtained, they are exchanged via electronic means between the proctoring institution and Athabasca University. Figure 24 shows the file sizes of images produced during the experiment activities. Typical Digital Subscriber Lines (DSL) reach transmission speeds in the range of 250 Kbps. Assuming this average rate, one can expect a 600 dpi colour image to take about 55 minutes to reach its destination.

Figure 24: File Size of Cover Works Captured By The Scanner



This section was used to present a summary of the data collected during the experiment. Its contents will be used to support the author’s conclusions outlined in the next section.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The results of the experiment were disappointing in that they showed that the AU Watermarking system cannot be used as a basis for the electronic exam exchange system described in the Methodology section of the paper. The data in the Technical Considerations section of the paper reveals that the watermark design used is too fragile for this purpose. The operations of routine marking, jpeg compression and transfer to paper all destroyed the embedded watermarks. More surprisingly, the Asprise OCR engine used could not reliably handle bitmap images of larger than the four character example used. In order to store the information noted in the Proposed Solution section, one would certainly require more than the maximum of 52 characters tested in the experiment. Furthermore, the files generated by the process are too large to use in a commercial setting. They would take up too much disk space and would take too long to transmit in any typical proctoring environment.

The algorithm used 8x8 pixel blocks to store binary bits of information. Although the DCT operation is not affected by image translation, the fact remains that any slight movement of the paper during scanning would result in the loss of a significant part portion of the pixels used for the watermarking process. This is likely the cause of the marks being completely destroyed after the transformation to paper. Modifications can also be introduced by scanner and printer drivers which can lead to adversely affecting the watermarks. A larger pixel block could have been considered but that would have

severely reduced the capacity of the cover works. The experiment's results have shown that documents were showing signs of saturation at the 200 dpi scanner setting so reducing the cover work's capacity was not a feasible option.

Another significant problem was that the watermarks themselves were too fragile. The OCR strategy was selected to provide robustness against errors. This robustness however is only realized in the data section of the image file. Any errors occurring in the bitmap header rendered the extracted watermark unusable.

For future study, there are a few approaches that can be looked at in further detail.

- Revising the algorithm to use an alternate means of achieving redundancy. OCR technology proved to be wasteful without providing much benefit. Over 16 000 KB of space was required to store a 52 byte ASCII character stream. Alternate redundancy techniques such as Hamming or Trellis codes are mentioned in the literary review section of the paper.
- Adjusting the algorithm so that it is smarter about where it embeds watermark information. The current implementation randomly selects blocks within the cover work for marking purposes. Given that the exam template can be made standard, ensuring that the watermark information is placed where students are not likely to write will decrease the probability of it being destroyed.

- Employing a different means of embedding the watermarks. Current researchers are experimenting with wavelet and Mellin-Fourier transform watermarking techniques. These alternate approaches might yield more encouraging results.
- Using JPEG files (or lower resolution BMP) files. The BMP watermark files generated in the experiment used 24 bits per pixel. JPEG representations of the same files used approximately 4% of the memory space.
- Storing only the data portion of the watermark images. This will likely improve the extraction rate. Results at the 600 dpi setting showed that if the file headers were not embedded in the watermarks, recognition rates would have been higher.

Although it is difficult to use watermarking for commercial paper based document systems, its utility in the general document management world should not be immediately discounted. The results have shown that if paper is not introduced into the process, positive results may be possible. With the emergence of Tablet PCs in the market place, it is conceivable, that students could handwrite their exams on these units without the exams even being transferred to paper.

The watermarking method investigated in this paper however is not robust enough to be used in a commercial exam exchange system. Furthermore, the experiment data collected showed that even if the watermarking system proposed was reliable, it would not be suitable for commercial use because of excessive processing time and image file sizes. Alternate approaches in this domain might yield more positive results.

REFERENCES

Antonacopoulos A., Karatzas D., Krawczyk H. & Wiszniewski B. (2004). The Lifecycle of a Digital Historical Document: Structure and Content. *DocEng '04*, pp.147-154

Barni M., & Bartolini F. (2004). *Watermarking Systems Engineering: Enabling Digital Assets Security And Other Applications*. Marcel Dekker

Baxes G., (1984). *Digital Image Processing: A Practical Primer*. Prentice-Hall

Boukong S., Toch B., Saad D., & Lowe D (2003). ICA for Watermarking Digital Images, *Journal of Machine Learning Research*, 4, pg 1471-1498

Chen Q., Yang X., & Zhao J. (2005). Robust Image Watermarking With Zernike Moments. *CCECE/CCGEI*, pp.1282-1285

Cole E. (2003). *Hiding In Plain Sight: Steganography And The Art of Covert Communication*. Wiley

Cox I., Miller M., & Bloom J. (2002). *Digital Watermarking*, Morgan Kaufmann Publishers.

Foley J., van Dam A., Feiner S. Hughes J., & Phillips R. (1994). *Introduction To Computer Graphics*. Addison-Wesley

Gatos B., Papamarkos N., & Chamzas C. (1997). A Binary-Tree Based OCR Technique for Machine-Printed Characters, *Engng. Applic. Artif. Intell*, 10, 4, pp.403-412

Gllavata J., Ewerth R., & Freisleben B. (2004). A Text Detection, Localization and Segmentation System for OCR In Images. *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering*.

Huang H., & Jain L.C. (2004). *Intelligent Watermarking Techniques*. World Scientific Publishing

Johnson N., & Jajodia S. (1998). Exploring Steganography: Seeing the Unseen, *IEEE Computer*, 0018,-9162 pg 26-34

Katzenbeisser S. & Petitcolas Fabian A.P. (2000). *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House.

Kipper G. (2004). *Investigator's Guide To Steganography*, CRC Press

Lambert S., & Lattin B. (2004). *Systems-Level View of Security Architectures, Practices and Standards*. Los Angeles, Lambert & Associates

Lin C., Wu M., Bloom J., Cox., Miller L., & Lui Y. (2000). Rotation, Scale, and Translation-resilient Public Watermarking for Images, *Security and Watermarking of Multimedia Contents*, SPIE-3971. pp.90–98

Liu Y. & Zhao J. (2004). Rotation, Scaling, Translation Invariant Image Watermarking Based on Radon Transform. *Proceedings of the First Canadian Conference on Computer and Robot Vision*.

Mori S., Nishida H. & Yamada H. (1999). *Optical Character Recognition*. Wiley

Mori S., Suen C. K., & Yamamoto K. (1992). Historical Review of OCR Research and Development. *Proceedings of the IEEE*,80,7, pp.1029-1058

Niblack W. (1986). *An Introduction To Digital Image Processing*. Prentice-Hall International

Nikolaidis N., & Pitas I. (1999). Digital Image Watermarking: An Overview. *International Conference on Multimedia Computing and Systems*.pp.1-6

ÓRuanaidh, Joseph J.K & Pun T (1997). Rotation, Scale and Translation Invariant Digital Watermarking. *Proceedings of the 1997 International Conference on Image Processing*.

ÓRuanaidh, Joseph J.K & Pun T (1998). Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing* 66(1998), pp.303-317

Pan J., Huang H. & Jain L.C (2004). *Intelligent Watermarking Techniques*. World Scientific Publishing

Parker J. R. (1997). *Algorithms For Image Processing And Computer Vision*. Wiley Computer Publishing

Russ J. C. (1992). *The Image Processing Handbook*. CRC Press

Sencar H., Ramkumar M., & Akansu A (2004). *Data Hiding Fundamentals and Applications*. London:Elsevier

Wayner P. (2002). *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*. Morgan Kaufmann

Wu V., Manmatha R., & Riseman E. M (1997). Finding Text In Images. *DL* 97, pp.3-12

Xin Y., Liao S. & Pawlak M. (2004). A Multibit Geometrically Robust Image Watermark Based on Zernike Moments, *Proceedings of the 17th International Conference on Pattern Recognition*.

Zhao J. & Koch E. (1995). Embedding Robust Labels Into Images For Copyright Protection. *Proc. of the Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*.

Zheng Y., Li H., & Doermann D. (2004). Machine Printed Text and Handwriting Identification in Noisy Document Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 3, pp.337-353

APPENDIX A

Basic Geometric Operations in Image Processing

Geometric Operation	Algebraic Equations
Moving (Translating) an object	$x' = x + d_x$ $y' = y + d_y$ <p>Moves the pixel identified by (x,y) in the vertical direction by d_y units and in the horizontal direction by d_x units.</p> <p>When applied to all pixels representing object, this operation moves the object in the image space.</p>
Scaling an object.	$x' = x \cdot s_x$ $y' = y \cdot s_y$ <p>Moves the pixel identified (x,y) either closer or farther away from the origin depending on the values of s_x and s_y. A scaling factor s, such that $0 < s < 1$ moves it closer to the axis in question. $s > 1$ moves it further from the axis. s_x moves the pixel relative to the x axis and s_y moves it relative to the y-axis.</p> <p>When applied to all pixels representing object, this operation moves the object in the image space and makes it larger or smaller. s_x and s_y must be equal for the object to shrink or expand by the same amount in both vertical and horizontal directions.</p>
Rotate an object	$x' = x \cdot \cos\theta - y \cdot \sin\theta$ $y' = x \cdot \sin\theta + y \cdot \cos\theta$ <p>Rotate the pixel about the origin by θ degrees in the counter clockwise direction.</p> <p>When applied to all pixels representing object, this operation rotates the object in the image space by θ degrees relative to the origin.</p>

APPENDIX B

OCR Recognition Techniques

In statistical recognition methods, features are defined as measurements on a glyph. The N measurements defined of a particular technique are combined to create an N dimensional vector. When classifying a normalized version of the glyph, some distance between its feature vector and the vectors for all known characters in the defined alphabet is calculated. This value is used to determine which alphabet character the glyphs examined is “closest” to.

Mori, Nishida et el (1999) present two examples of statistical methods in their work. The first is called the cross-correlation method. The feature being looked at in this case is a projection of the glyph in the domain of the image defined by $f(x,y)/|f|$ for all x and y in the domain. The function f represents the glyph its image space and $|f|$ is the magnitude (or norm) of f. The cosine of the angle between f and known alphabet glyph g can be found when the projection of $f(x,y)/|f|$ on $g(x,y)/|g|$ is calculated. The following equation captures this relationship.

Figure 25: Projection Equation

$$\cos \theta = \frac{f \bullet g}{|f| |g|}$$

where $f \bullet g$ is the inner product of f and g

This leads to the following equation used to implement the cross-correlation method.

Figure 26: Equation For Cross-Correlation

$$S(f) = \frac{\iint_R f(x,y)g(x,y)dxdy}{\sqrt{\iint_R f(x,y)^2 dxdy} \sqrt{\iint_R g(x,y)^2 dxdy}}$$

where R is the domain of the glyph's image space

$f(x,y)$ is a normalized representation of the glyph being analysed and is equal to 1 if it represents a text character, 0 otherwise

$g(x,y)$ is a glyph of a known character and is equal to 1 if it represents a text character, 0 otherwise

The function S obeys the relationship $0 \leq S \leq 1$ since all values of f and g are positive.

As $f(x,y)$ approaches $g(x,y)$ the function S approaches 1 so S is dependent on the

Euclidean distance between f and g. In order to consider the glyph represented by f to be considered the character represented by g, S(f) must exceed a defined threshold and be the

maximum value of $S(f)$ for all possible glyphs g in the alphabet. The cross-correlation method is an example of a global threshold technique. In these types of methods character classification decisions are made based on information from the whole glyph. Global methods provide high recognition rates.

The second technique presented by Mori, Nishida et al (1999) is based on Boolean algebra. Assuming that the glyph being analysed has been normalized, a sample of pixels can be analysed and compared with the glyphs of known alphabet characters in question. The following equation provides a solution to the logical method.

For the given character A ,

$$A = W \cap B$$

where

$$W = \left(\sum^n f(i, j) g^w(i, j) \leq T_w \right)$$

$$B = \left(\sum^m f(i, j) g^b(i, j) \geq T_b \right)$$

T_B and T_W are threshold values for text and background pixels respectively.

$f(i,j)$ is a function which represents the glyph being evaluated. It is equal to 1 if the pixel in row i and column j in the glyph is a text pixel, -1 otherwise.

$g^w(i,j)$ is a frame representing the background pixels in the glyph for character A. The pixel at row i , column j will be 1 if that position is a background pixel, 0 otherwise

$g^b(i,j)$ is a frame representing the text pixels in the glyph for character A. The pixel at row i , column j will be 1 if that position is a text pixel, 0 otherwise

n and m are the cardinality of the sample background and text sets respectively.

Figure 27: Recognition Based On Boolean Algebra

W5		B5	W4	
	B4			
B1				B4
		W1		
B2				
			B3	W3
	W2			

The number and actual location of the pixels being checked are strategically selected in order to increase accuracy. The example in Figure 27 illustrates how these sets could be selected.

The logical method is a local threshold method. It uses a limited amount of information in the glyph to make recognition decisions. Although these methods tend to be less accurate than global methods, they are also less computationally expensive. The efficiency of these algorithms can be improved by stopping checks once it can be determined that the recognition threshold defined cannot be obtained.

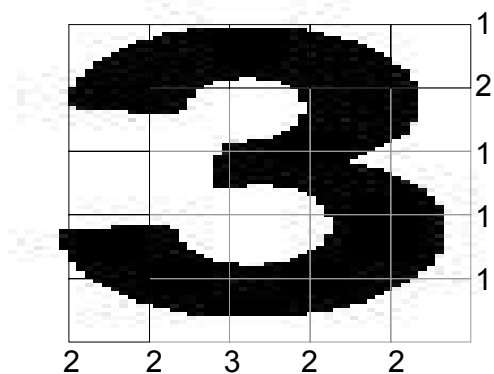
In structural techniques, features are defined to be small components. Any known character is defined to be a combination of these glyphs. A glyph's features are compared with the features of known alphabet glyphs to make recognition decisions. In Gatos, Papamarkoes et al's Binary-Tree Based OCR Technique (1997), features are defined to be attributes such as a hole, an endpoint or a text to background transition point. These types of features are independent of scaling or rotation. This eliminates the need to normalize glyphs or rotate them to compensate for skewed text. A simple structural technique will now be discussed.

Mori, Nishida and Yamada (1999) describe a structural method developed by Weeks where features are defined as the number of black (text) regions on a line which passes

through a glyph. This feature is calculated on equally spaced, parallel lines in each of four directions. The directions are vertical, horizontal and two orthogonal diagonals.

Figure 28 gives an example of such analysis. For simplicity, the diagonal values are not shown.

Figure 28: An Example Of Weeks's Recognition Analysis



APPENDIX C

Cryptography

Cryptographic tools ensure that sensitive information remains secret by converting such data into a form that is unrecognizable by unauthorized parties. Data in this form is referred to as cipher text. Individuals that are allowed to access the information convert it back to its original, readable format (known as plain text) using a decryption algorithm and a key. This section will describe various types of encryption schemes currently in use.

Symmetric key encryption is the simplest type of encryption scheme. They have a single key that is used for both the encryption and decryption algorithm. This key should only be known by the sending party and the party or parties that it intends to send protected data to. The actual implementation of symmetric encryption typically involves one or a combination of the following techniques. (Cole, 2003)

- Substitution – The generated key creates a one to one mapping between characters in the plain text alphabet with those in the cipher text alphabet.
- Permutation – The generated key defines how the characters in plain text data should be scrambled
- Exclusive-OR – The generated key is XOR'ed with the plain text to produce the cipher text. This requires that both the key and the original data can be represented in binary form. The XOR operation takes the corresponding bits of

two equal length binary numbers and produces another binary number with the following properties. If the bits from the two numbers have the same value, the corresponding bit in the output would be zero. Otherwise, the output bit is one.

Using symmetric key systems provide the advantage of being fast relative to other models. Disadvantages include the inability to achieve non-repudiation and the requirement that the key must be transported to the receiver securely. Schemes of this nature also do not scale well. In a system where n users must securely communicate with each other, $n(n-1)/2$ keys must remain secure.

Asymmetric key encryption schemes require a key pair to be generated. If one key is used to encrypt data, the other key is the only one that can be used to decrypt it. In practice, these are commonly known as public/private keypairs. Each party wishing to communicate has a public key that is distributed freely. It also retains its own private key that no one else has privy to. In order to send encrypted data, one uses the receiver's public key during the encryption process. The cryptographic system guarantees that receiver's corresponding private key is the only one that will successfully decrypt the cipher text data given the appropriate decryption algorithm. Since the receiver's private key is supposed to be only known by the receiver itself, the data remains private. The sender encrypting data using its private key can achieve non-repudiation. Any user with the sender's public key can decrypt the data. Successful decryption using the public key

implies that the associated private key must have been used to encrypt it. Since no other party knows its private key, the sender must have encrypted the cipher text.

Hash encryption techniques transform a piece of information into a fixed length value. Unlike the previously mentioned approaches, there is no way to obtain the original data from the output. Hashing is commonly used to add the non-repudiation property to a system. Hashing functions are designed to virtually guarantee that unique inputs yield unique output values. Given a hashing function H and two pieces of information, x_1 and x_2 , it is statically unlikely that $H(x_1) = H(x_2)$ for all possible x_1 and x_2 . Furthermore, if one has x_1 and $H(x_1)$, it is computationally infeasible to find a second piece of information q such that $H(x_1) = H(q)$ even if one does exist. This technique is used to create digital signatures for files.

APPENDIX D

Watermarking System User Guide

The command line application built for this research project provides the end user with the ability to embed watermarks into a cover work and to extract images from a cover work.

Build Considerations

The C++ code provided with this work was compiled in the Visual Studio 2005 development environment using the Visual C++ 8.0 compiler. Once this development environment is installed, opening the AUCmdLineWatermarkApp.sln solution file will cause the relevant source files and project to become accessible through the Visual Studio IDE.

Dependencies

The GDI+ library and the C Runtime Library from Microsoft are required for the application to run correctly.

Command Line Parameters

For the embedding function, the following syntax holds:

AUCmdLineWatermarkApp /embed /infile *inimg* /outfile *outimg* /msg *mark* /markfile
markimg

- *inimg* – the filename of the original work to be marked
- *outimg* – the filename of the cover work that is created
- *mark* – the ASCII representation of the watermark to be embedded
- *markfile* – the watermark embedded into the cover work. It is the bitmap representation of *mark*

For the extraction function, the following syntax holds:

AUCmdLineWatermarkApp /extract /infile *inimg* /outfile *outimg* /msg *mark* /markfile
markimg

- *inimg* – the filename of the covermark potentially containing a watermark
- *outimg* – the filename of the watermark extracted from the cover work