Electronic Marking and Identification Techniques to Discourage Document Copying

Jack T. Brassil, *Senior Member, IEEE*, Steven Low, *Member, IEEE*, Nicholas F. Maxemchuk, *Fellow, IEEE*, and Lawrence O'Gorman, *Senior Member, IEEE*

Abstract— **Modern computer networks make it possible to distribute documents quickly and economically by electronic means rather than by conventional paper means. However, the widespread adoption of electronic distribution of copyrighted material is currently impeded by the ease of unauthorized copying and dissemination. In this paper we propose techniques that discourage unauthorized distribution by embedding each document with a unique codeword. Our encoding techniques are indiscernible by readers, yet enable us to identify the sanctioned recipient of a document by examination of a recovered document. We propose three coding methods, describe one in detail, and present experimental results showing that our identification techniques are highly reliable, even after documents have been photocopied.**

I. Introduction

ELECTRONIC distribution of publications is increasingly available through on-line text databases, CD-ROM's, computer network based retrieval services, and electronic libraries [1]–[6]. One electronic library, the RightPages[1] Service [7]–[9], has been in place within Bell Laboratories since 1991, and has recently been installed at the University of California in San Francisco. Electronic publishing is being driven by the decreasing cost of computer processing and high quality printers and displays. Furthermore, the increased availability of low cost, high speed data communications makes it possible to distribute electronic documents to large groups quickly and inexpensively [10].

While photocopy infringements of copyright have always concerned publishers, the need for document security is much greater for electronic document distribution [11], [12]. The same advances that make electronic publishing and distribution of documents feasible also increase the threat of "bootlegged" copies. With far less effort than it takes to copy a paper document and mail it to a single person, an electronic document can be sent to a large group by electronic mail. In addition, while originals and photocopies of a paper document can look and feel different, copies of electronic documents are identical.

In order for electronic publishing to become accepted, publishers must be assured that revenues will not be lost due to theft of copyrighted materials. Widespread unauthorized document dissemination should ideally be at least as costly or difficult as obtaining the documents legitimately. Here we define "unauthorized dissemination" as distribution of documents without the knowledge of—and payment to—the publisher; this contrasts legitimate document distribution by the publisher or the publisher's electronic document distributor. This paper describes a means of discouraging unauthorized copying and dissemination. A document is marked in an indiscernible way by a codeword identifying the registered owner to whom the document is sent [13]. If a document copy is found that is suspected to have been disseminated without authorization, that copy can be decoded and the registered owner identified.

The techniques we describe here are complementary to the security practices that can be applied to the legitimate distribution of documents. For example, a document can be encrypted prior to transmission across a computer network [14], [15]. Then even if the document file is intercepted or stolen from a database, it remains unreadable to those not possessing the decrypting key. The techniques we describe in this paper provide security *after* a document has been decrypted, and is thus readable to all.

In addition to discouraging unauthorized dissemination of documents distributed by computer network, our proposed encoding techniques can also make paper copies of documents traceable. In particular, the codeword embedded in each document survives plain paper copying. Hence, our techniques can also be applied to "closely held" documents, such as confidential, limited distribution correspondence. We describe this both as a potential application of the methods and an illustration of their robustness in noise.

II. Document Coding Methods

Document marking can be achieved by altering the text formatting, or by altering certain characteristics of textual elements (e.g., characters). The goal in the design of coding methods is to develop alterations that are reliably decodable (even in the presence of noise) yet largely indiscernible to the reader. These criteria, reliable decoding and minimum visible change, are somewhat conflicting; herein lies the challenge in designing document marking techniques.

The marking techniques we describe can be applied to either an image representation of the document or to a document format file. The document format file is a computer

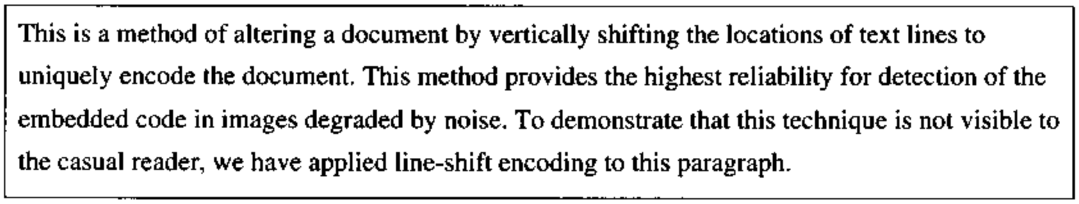
Manuscript received August 8, 1994; revised March 1, 1995. A preliminary version of this paper was presented at IEEE INFOCOM'94.

The authors are with the AT&T Bell Laboratories, Murray Hill, NJ 07974 USA.

IEEE Log Number 9413489.

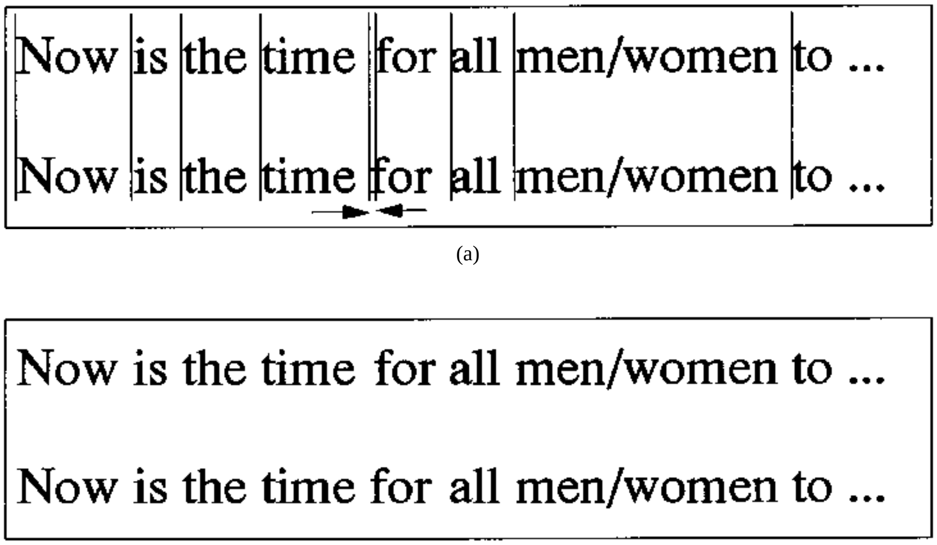
[1] RightPages is a trademark of AT&T.

0733–8716/95$04.00 © 1995 IEEE



This is a method of altering a document by vertically shifting the locations of text lines to uniquely encode the document. This method provides the highest reliability for detection of the embedded code in images degraded by noise. To demonstrate that this technique is not visible to the casual reader, we have applied line-shift encoding to this paragraph.

Fig. 1. Example of line-shift coding. The second line has been shifted up by $1/300$ inch.



Now is the time for all men/women to ...

Now is the time for all men/women to ...

(a)

Now is the time for all men/women to ...

Now is the time for all men/women to ...

(b)

Fig. 2. Example of word-shift coding. In (a), the top text line has added spacing before the "for," the bottom text line has the same spacing after the "for." In (b), these same text lines are shown again without the vertical lines to demonstrate that either spacing appears natural.

file describing the document content and page layout (or formatting), using standard format description languages such as PostScript,[2] TeX, troff, etc. It is from this format file that the image—what the reader sees—is generated. The image representation describes each page (or subpage) of a document as an array of pixels. The image may be bitmap (also called binary or black-and-white), gray-scale, or color. For this work, we describe both document format file and image coding techniques, however we restrict the latter to bitmaps encoded within the binary-valued text regions.

Common to each technique is that a codeword is embedded in the document by altering particular textual features. For instance, consider the codeword 1101 (binary). Reading this code right to left from the least significant bit, the first document feature is altered for bit 1, the second feature is not altered for bit 0, and the next two features are altered for the two 1 bits. It is the type of feature that distinguishes each particular encoding method. We describe these features for each method below and give a simple comparison of the relative advantages and disadvantages of each technique.

The three coding techniques that we propose illustrate different approaches rather than form an exhaustive list of document marking techniques. The techniques can be used either separately or jointly. Each technique enjoys certain advantages or applicability as we discuss below.

A. Line-Shift Coding

This is a method of altering a document by vertically shifting the locations of text lines to encode the document uniquely. This encoding may be applied either to the format file or to the bitmap of a page image. The embedded codeword may be extracted from the format file or bitmap. In certain cases this decoding can be accomplished without need of the original image, since the original is known to have uniform line spacing (i.e., "leading") between adjacent lines within a paragraph.

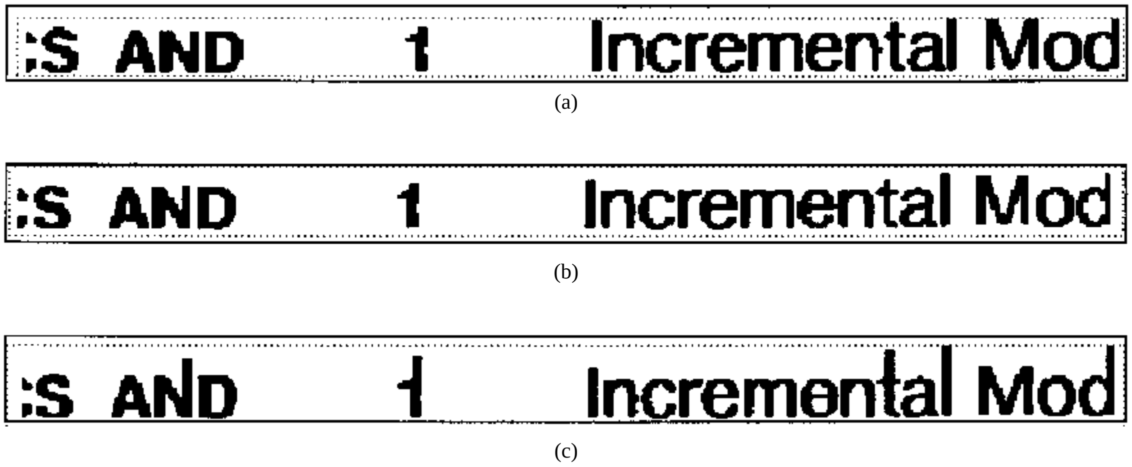
B. Word-Shift Coding

This is a method of altering a document by horizontally shifting the locations of words within text lines to encode the document uniquely. This encoding can be applied to either the format file or to the bitmap of a page image. Decoding may be performed from the format file or bitmap. The method is least visible when applied to documents with variable spacing between adjacent words. Variable spacing in text documents is commonly used to distribute white space when justifying text.

Because of this variable spacing, decoding requires the original image—or more specifically, the spacing between words in the unencoded document. See Fig. 2 for an example of word-shift coding.

Consider the following example of how a document might be encoded with word-shifting. For each text line, the largest

[2] PostScript is a trademark of Adobe Systems, Inc.



(a)

(b)

(c)

Fig. 3. Example shows feature coding performed on a portion of text from a journal table of contents. In (a), no coding has been applied. In (b), feature coding has been applied to select characters. In (c), the feature coding has been exaggerated to show feature alterations.

and smallest spacings between words are found. To code a line, the largest spacing is decremented by some amount and the smallest is augmented by the same amount. This maintains the text line length, and produces little qualitative change to the text image.

C. Feature Coding

This is a coding method that is applied either to a format file or to a bitmap image of a document. The image is examined for chosen text features, and those features are altered, or not altered, depending on the codeword. Decoding requires the original image, or more specifically, a specification of the change in pixels at a feature. There are many possible choices of text features; here, we choose to alter upward, vertical endlines—that is the tops of letters, *b*, *d*, *h*, etc. These endlines are altered by extending or shortening their lengths by one (or more) pixels, but otherwise not changing the endline feature. See Fig. 3 for an example of feature coding.

Among the proposed encoding techniques, line-shifting is likely to be the most easily discernible by readers. However we also expect line-shifting to be the most robust type of encoding in the presence of noise. This is because the long lengths of text lines provide a relatively easily detectable feature. For this reason, line shifting is particularly well suited to marking documents to be distributed in paper form, where noise can be introduced in printing and photocopying. As we will show in Section III, our experiments indicate that we can easily encode documents with line shifts that are sufficiently small that they are not noticed by the casual reader, while still retaining the ability to decode reliably.

We expect that word-shifting will be less discernible to the reader than line-shifting, since the spacing between adjacent words on a line is often varied to support text justification. Feature encoding can accommodate a particularly large number of sanctioned document recipients, since there are frequently two or more features available for encoding in each word. Feature alterations are also largely indiscernible to readers. Feature encoding also has the additional advantage that it can be applied simply to image files, which allows encoding to be introduced in the absence of a format file.

Implementing any of the three document marking techniques described above incurs certain "costs" for the electronic document distributor. While the exact nature of the costs is implementation dependent, we can nonetheless make several general remarks based on our experience [16]. Distributors must incur a small penalty in maintaining a library of "codebooks" which contain a mapping of embedded codewords and recipients for each original (unmarked) document they mark and distribute. A larger penalty is paid in distributing images rather than higher level page descriptions—roughly 3–5 times the number of bits must be transmitted to the subscriber.[3]

A technically sophisticated "attacker" can detect that a document has been encoded by any of the three techniques we have introduced. Such an attacker can also attempt to remove the encoding (e.g., produce an unencoded document copy). Our goal in the design of encoding techniques is to make successful attacks extremely difficult or costly. We will return to a discussion of the difficulty of defeating each of our encoding techniques in Section IV.

III. Implementation and Experimental Results for Line-Shift Coding Method

In this section we describe in detail the methods for coding and decoding we used for testing the line-shift coding method. Each intended document recipient was preassigned a unique codeword. Each codeword specified a set of text lines to be moved in the document specifically for that recipient. The length of each codeword equaled the maximum number of lines that were displaced in the area to be encoded. In our line-shift encoder, each codeword element belonged to the alphabet $\{-1, +1, 0\}$, corresponding to a line to be shifted up, down or remain unmoved.

Though our encoder was capable of shifting an arbitrary text line either up or down, we found that the decoding

[3] For a technical journal article, a compressed, 300 dpi bitmap representation is nominally 3–5 times larger than a compressed page description language representation.

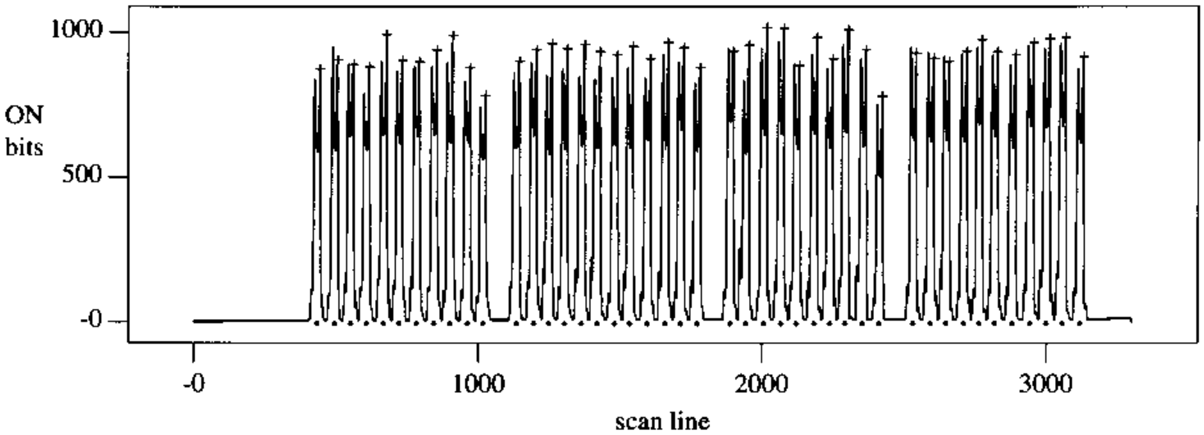


Fig. 4. Profile of a recovered document page. Decoding a page with line shifting requires measuring the distances between adjacent text line centroids (marked with •) or baselines (marked with +) and deciding whether white space has been added or subtracted.

performance was greatly improved by constraining the set of lines moved. In the results presented in this paper, we used a *differential* (or difference) encoding technique. With this coding we kept every other line of text in each paragraph unmoved, starting with the first line of each paragraph. Each line between two unmoved lines was always moved either up or down. That is, for each paragraph, the 1st, 3rd, 5th, etc. lines were unmoved, while the 2nd, 4th, etc. lines were moved. This encoding was partially motivated by image defects we will discuss later in this section. Note that the consequence of using differential encoding is that the length of each codeword is cut approximately in half. While this reduces the potential number of recipients for an encoded document, the number can still be extremely large. In each of our experiments we displaced at least 19 lines, which corresponds to a potential of at least $2^{19} = 524,288$ distinct codewords/page. More than a single page per document can be coded for a larger number of codeword possibilities or redundancy for error-correction.

Each of our experiments began with a paper copy of an encoded page. Decoding from the paper copy first required scanning to obtain the digital image. Subsequent image processing improved detectability; salt-and-pepper noise was removed [17] and the image was deskewed to obtain horizontal text [18], [19]. Text lines were located using a horizontal *projection profile*. This is a plot of the summation of ON-valued pixels along each row. For a document whose text lines span horizontally, this profile has peaks whose widths are equal to the character height and valleys whose widths are equal to the white space between adjacent text lines. The distances between profile peaks are the interline spaces.

The line-shift decoder measured the distance between each pair of adjacent text line profiles (within the page profile). This was done by one of two approaches—either we measured the distance between the baselines of adjacent line profiles, or we measured the difference between centroids of adjacent line profiles. A baseline is the logical horizontal line on which characters sit; a centroid is the center of mass of a text line profile. As seen in Fig. 4, each text line produces a distinctive profile with two peaks, corresponding to the midline and baseline. The peak in the profile nearest the bottom of each text line is taken to be the baseline. To define the centroid of a text line precisely, suppose the text line profile runs from scan line y, $y+1$, $\cdots$, to $y+w$, and the respective number of ON bits/scan line are $h(y)$, $h(y+1)$, $\cdots$, $h(y+w)$. Then the text line centroid is given by

$$\frac{yh(y) + \cdots + (y+w)h(y+w)}{h(y) + \cdots + h(y+w)}. \tag{3.1}$$

The measured interline spacings (i.e., between adjacent centroids or baselines) were used to determine if white space has been added or subtracted because of a text line shift. This process, repeated for every line, determined the codeword of the document—this uniquely determined the original recipient.

We now describe our decision rules for detection of line shifting in a page with differential encoding. Suppose text lines $i-1$ and $i+1$ are not shifted and text line i is either shifted up or down. In the unaltered document, the distance between adjacent baselines, or baseline spacings, are the same. Let s_{i-1} and s_i be the distances between baselines $i-1$ and i, and between baselines i and $i+1$, respectively, in the altered document. Then the *baseline detection decision rule* is:

$$\begin{array}{ll} if\ s_{i-1} > s_i: & decide\ line\ i\ shifted\ down \\ if\ s_{i-1} < s_i: & decide\ line\ i\ shifted\ up \\ otherwise\ \ : & uncertain. \end{array} \tag{3.2}$$

Unlike baseline spacings, centroid spacings between adjacent text lines in the original unaltered document are not necessarily uniformly spaced. In centroid-based detection, the decision is based on the difference of centroid spacings in the altered and unaltered documents. More specifically, let s_{i-1} and s_i be the centroid spacings between lines $i-1$ and i, and between lines i and $i+1$, respectively, in the altered document; let t_{i-1} and t_i be the corresponding centroid spacings in the unaltered document. Then the *centroid detection decision rule* is:

$$\begin{array}{ll} if\ s_{i-1} - t_{i-1} > s_i - t_i: & decide\ line\ i\ shifted\ down \\ otherwise\ \ : & decide\ line\ i\ shifted\ up. \end{array} \tag{3.3}$$

An *error* is said to occur if our decoder decides that a text line was moved up (down) when it was moved down (up). In baseline detection, a second type of error exists. We say that the decoder is *uncertain* if it cannot determine whether a line was moved up or down. Since, for our encoding method, every other line is moved and this information is known to the decoder, false alarms do not occur.

A. Experimental Results for Line-Shift Coding

We conducted two sets of experiments. The first set tested how well line-shift coding works with different font sizes and different line spacing shifts in the presence of limited, but typical, image noise. The second set tested how well a fixed line spacing shift could be detected as document degradation became increasingly severe. In this section, we first describe these experiments and then present our results.

The equipment we used in both experiments was as follows: a Ricoh FS1S 400 dpi Flat Bed Electronic Scanner, Apple LaserWriter IIntx 300 dpi laser printer, and a Xerox 5052 plain paper copier.[4] The printer and copier were selected in part because they are typical of equipment found in wide use in office environments. The particular machines we used could be characterized as being heavily used but well maintained.

Writing the software routine to implement a rudimentary line-shift encoder for a PostScript input file was simple. We chose the PostScript format because: 1) it is the most common Page Description Language in use today, 2) it enables us to have sufficiently fine control of text placement, and 3) it permits us to encode documents produced by a wide variety of word processing applications. PostScript describes the document content a page at a time. Roughly speaking, it specifies the content of a text line (or text line fragment such as a phrase, word, or character) and identifies the location for the text to be displayed. Text location is specified by an x-y coordinate representing a position on a virtual page; this position can typically be altered by arbitrarily small displacements. However, most personal laser printers in common use today have a 300 dpi "resolution," so they are unable to distinctly render text subject to a displacement of less than 1/300 inch.

1) Variable Font Size Experiment: The first set of experiments each used a single-spaced page of text in the Times-Roman font. The page was coded using the differential encoding scheme. We performed nine experiments using font sizes of 8, 10, or 12 points and shifting alternate lines (within each paragraph) up or down by 1, 2, or 3 pixels. Each page of 8, 10, and 12 point size text extended for 23, 21, and 19 lines, respectively. Different numbers of encoded lines per page arise naturally, since as the font size decreases, more lines can be placed on the page, permitting more information to be encoded. Since our printer has a 300 dpi resolution, each pixel corresponds to $1/300$ inch, or approximately one-quarter of a printer's "point." Each coded page was printed on the laser printer, then copied three times. We will refer to the laser printed page as the 0th copy; the nth copy, $n \geq 1$, is produced by copying the $n-1$st copy. The third copy was then decoded to extract the codeword. That is, we electronically scanned the third copy, processed the bitmap image to generate the profile, processed the profile to generate the text line spacings (both baseline and centroid spacings), and detected the codeword using these measurements and rules (3.2)–(3.3).

The results of the variable font size experiment were extremely good for all cases. Using the centroid detection method, all line shifts were successfully detected without error. Using the baseline detection method, all line shifts for the 10 point font size were successfully detected without error. All line shifts of 2 and 3 pixels were also detected without error for the 8 and 12 point size cases. For 8 point size text with 1 pixel spacing, 18 of 23 line shifts were correctly detected, though the remaining 5 line shifts were deemed *uncertain*. For 12 point size text with 1 pixel spacing, 18 of 19 line shifts were correctly detected, while 1 line shift was incorrectly detected (i.e., 1 error). In summary, both baseline and centroid approaches detected without error for spacings of at least 2 pixels; the centroid approach also had no errors for a 1 pixel spacing.

Though it is not apparent from the results we have stated, it is noteworthy that some variability will occur in the detection performance results, even in repeated "decoding" of the same recovered page. This variability is due in part to randomness introduced in electronic scanning. If a page is scanned several times, different skew angles will ordinarily occur. The skew will be corrected slightly differently in each case, causing detection results to vary.

To illustrate this phenomena, we rescanned in the test case (8 point text, 1 pixel spacing) 3 additional times. The initial text line skew angle (i.e., before deskewing) differed for each scan. In the three rescans, we observed the following decoding results under baseline detection: 5 uncertain, 3 uncertain and 1 error, and 6 uncertain. The line spacings that could not be detected or were in error varied somewhat across the retries. This suggests that there may be some decoding performance gained by scanning a single page multiple times, and combining the results (e.g., averaging).

2) Plain Paper Copying Experiment: For the second set of experiments, we encoded a single-spaced page of text using differential encoding. We used a 10 point Times-Roman font, and a 1 pixel line shift. Twenty-one lines were shifted on the page. We then made repeated copies (the 1st, ..., 10th copy) of the page, and used each copy in a separate experiment. Hence, each successive experiment used a slightly more degraded version of the same text page.

Detection results were surprisingly good. The centroid detection method successfully detected all 21 line shifts for each generation of photocopy (through the 10th copy). The baseline detection method successfully detected all lines on every copy generation, with the following exceptions: 1 error was made on the 4th, 5th, 6th, 7th, and 10th copy, 2 errors on the 9th copy; 1 uncertain on the 3rd, 4th, and 10th copy, 2 uncertains on the 7th copy, and 4 uncertains on the 8th copy. In summary, the baseline detection method successfully detected at least 16 line shifts on each copy generation.

Though further testing must be done to understand better how the coding is affected by noise, our results indicate that, for baseline decoding, detection errors and uncertainties do not increase monotonically with the number of copies. Further, the line spacings that could not be detected correctly varied

[4] Xerox and 5052 are trademarks of Xerox Corp. Apple and LaserWriter are trademarks of Apple Computer, Inc. Ricoh and FS1 are trademarks of Ricoh Corp.

somewhat from copy to copy. This suggests that line spacing "information" is still present in the text baselines, and can perhaps be made available with some additional processing.

We have reported the uncoded error performance of our marking scheme. But the 21 line shifts used in the experiment were not chosen arbitrarily. The 21 line shifts comprised 3 concatenated codewords selected from a Hamming $(7,4)$ block code, a 1-error correcting code. Had we chosen to make use of this error correction, roughly each third of a page would have been protected from 1 error. Many, but not all, of the baseline decoding errors and uncertainties would have been corrected by this encoding. Of course, using an error-correcting code would require increasing the number of line shifts to produce the same maximum number of uniquely encoded documents. We expect to use error correction to increase detection performance in future experiments, particularly those where text displacements are smaller than those we have considered here. We also expect that interleaving codewords across one or more pages will improve overall detection performance.

Our experimental results reveal that centroid-based detection outperforms baseline-based detection for pages encoded with small line shifts (i.e., 1 pixel) and subject to large distortion. This performance difference arises largely because baseline locations are integer-valued, while centroid locations, being averages, are real-valued. Since baseline locations are determined by detection of a peak in the text line profile, this location can be of low accuracy when the peak is not sharp due to some page skew, noise, a short text line, etc. A single scan line error in locating a text baseline is sufficient to introduce a detection error when text lines are encoded with a 1 pixel shift.

Though the use of centroids is less subject to certain imaging defects than are baselines, baseline coding provides other benefits. In particular, encoded documents can be decoded without reference to the original, unaltered document. A secure document distributor would then be relieved of the need to maintain a library of original document centroid spacings for decoding. Of course, both detection techniques can be used jointly to provide a particularly robust, low error probability detection scheme.

B. Discussion and Implications of Image Defects

Image defects [20], [21] resulting from plain paper copying are all too familiar to the reader. We now briefly discuss the defects most significantly affecting our detection results. Our discussion is largely qualitative—a more quantitative discussion of image defects and their physical underpinnings is beyond the scope of this paper.

The primary troublesome defect we encountered was text line skew, or the rotation of text lines about a point. In most experiments we observed skew angles between $[-3°, +3°]$. Text line skew was largely removed by image rotation, at the expense of the introduction of some distortion due to bilinear interpolation of sampled data.

Blurring (i.e., edge raggedness) also increased with the number of copies produced. However, blurring seemed to have surprisingly minor implications in detection performance. It is possible that blurring introduces noise in a symmetrical fashion on text lines, so it does not contribute significantly to displacing centroid locations. Plain paper copies were produced at the copier's nominal "copy darkness" setting; blurring typically increases with copy darkness. As the number of copies increased, copy darkness generally varied over a page; regions of severe fading were sometimes observed. It is unclear whether blurring or fading is more detrimental to decoding performance.

Expansion or shrinking of copy size is another potential problem. It is not unusual to discover a 4% page length or width change after 10 copies. Further, expansion along the length and width of a page can be markedly different. Copy size changes forced us to use differential encoding—that is, encoding information in the relative rather than absolute shifts between adjacent text lines.

C. A Noise Model

In this subsection we present a simple model of the noise affecting text line centroids. We distinguish two types of noise. The first type of noise models the distortion in printing and scanning the document; the second type models the distortion in copying. This second type of noise increases with the number of copies while the first type does not.

An unaltered page of text with $n+1$ text lines yields $n+1$ vertical coordinates $y_1, \cdots, y_{n+1}$, that represent the centroids of the text lines, measured from, say, the top page margin. The centroid spacings, or distance in scan lines between adjacent centroids, are given by

$$t_i = y_{i+1} - y_i \qquad i = 1, \cdots, n. \tag{4.1}$$

Hence, for detecting line-shifts, a page of $n+1$ text lines is effectively described by n centroid spacings.

The ith line spacing shift c_i is positive if extra space has been added, negative if space has been subtracted, and zero otherwise. This line spacing shift changes the original ith centroid spacing from t_i to $t_i + c_i$. Let s_i^j be the ith centroid spacing in the jth copy of an altered document. The printer noise, ν_i, models the cumulative effect (on the ith centroid spacing) of distortion introduced by printing, scanning, and image processing. We assume that the printer noise ν_i is strictly additive and logically distorts the centroid spacings of the original paper copy to

$$s_i^0 = t_i + c_i + \nu_i, \qquad i = 1, \cdots, n, \tag{4.2}$$

where ν_i, $i = 1, \cdots, n$, are independent and identically distributed Gaussian random variables. This assumption is supported by our measurements [22], which yield a mean of $\mu_1 = 0.0528\ pixel$ and variance of $\sigma_1^2 = 0.140\ pixel^2$.

Let N_i^j be the random noise that summarizes the cumulative effect of skewing, scaling, and other photographic distortions introduced on the ith centroid spacing s_i^{j-1} by making the jth copy. Then the centroid spacings on the jth copy are

$$s_i^j = s_i^{j-1} + N_i^j, \qquad i = 1, \cdots, n; j = 1, \cdots, K. \tag{4.3}$$

Hence, the centroid spacing s_i^j is corrupted by the cumulative noise:

$$s_i^j = s_i^0 + \left(N_i^1 + \cdots + N_i^j\right). \tag{4.4}$$

Since the physical processes of printing, scanning, and image processing are independent of copying, we will assume that the random variables ν_i, $i = 1, \cdots, n$, are independent of N^j_i, $i = 1, \cdots, n$; $j = 1, \cdots, K$. Our measurements suggest a surprisingly simple statistical behavior for the random copier noise. The noise components N^j_i, $j = 1, 2, \cdots, K$, are well approximated by i.i.d. Gaussian random variables with mean $\mu_2 = 0.066\ pixel$ and variance $\sigma^2_2 = 0.017 pixel^2$. Hence, the centroid spacing s^j_i on the jth copy is

$$s^j_i = s^0_i + \eta^j_i, \qquad i = 1, \cdots, n, \tag{4.5}$$

where η^j_i is Gaussian with mean $j\mu_2$ and variance $j\sigma^2_2$.

We now combine printer noise and copier noise to estimate the "bit" error probability under centroid detection. Consider three adjacent, differentially encoded text lines labeled such that lines $i-1$ and $i+1$ are unshifted while line i is shifted (up or down) by $|c|$ pixels. The corresponding centroid spacings s^j_{i-1} and s^j_i on the jth copy are

$$s^j_{i-1} = t_{i-1} + c + \nu_{i-1} + \eta^j_{i-1} \tag{4.6}$$

$$s^j_i = t_i - c + \nu_i + \eta^j_i, \tag{4.7}$$

where η^j_i are defined as in (4.5).

Next define the decision variable $D \stackrel{\Delta}{=} (\nu_{i-1} - \nu_i) + \left(\eta^j_{i-1} - \eta^j_i\right)$. Since the random variables $\nu_{i-1}, \nu_i, \eta^j_{i-1}, \eta^j_i$ are mutually independent, the variable D is Gaussian with zero mean and variance $2\left(\sigma^2_1 + j\sigma^2_2\right)$.

Now suppose the jth copy of the document is recovered and is to be decoded. Applying (4.6)–(4.7) to the centroid detection decision rule (3.3) and simplifying yields

$$\begin{array}{ll} if\ D > -2c: & decide\ line\ i\ shifted\ down \\ otherwise\ \ : & decide\ line\ i\ shifted\ up. \end{array} \tag{4.8}$$

Hence, the probability that a given line shifted by 1 pixel is decoded in error is

$$\begin{aligned} & p(decide\ up\ shift|down\ shift)p(down\ shift)+ \\ & p(decide\ down\ shift|up\ shift)p(up\ shift) \qquad (4.9) \\ =& p(D \leq -2)p(down\ shift) + p(D > 2)p(up\ shift) \\ =& p(D > 2). \qquad (4.10) \end{aligned}$$

The error probability is easily evaluated using the complementary error function. Using the measurements $\sigma^2_1 = 0.140\ pixel^2$ and $\sigma^2_2 = 0.017\ pixel^2$, the probability that a 1 pixel line shift is decoded in error is only approximately 2% on the 20th copy.

IV. Detecting and Defeating Image Encoding

It appears that all document coding schemes, including the ones introduced in this paper, can be detected and defeated. Successful attacks on encoded documents arguably involve a degree of technical sophistication and effort. The sophistication of successful attacks can vary, introducing various tradeoffs. For example, document presentation quality may be sacrificed by the process of removing an encoding. An extreme case is for an attacker simply to obliterate an encoding by adding enough noise to render an image undecodeable, however, this may also render the document illegible or marginally legible.

Hence our objectives in designing attack-resistant image coding schemes are ideally to

1) ensure that substantial effort is required to a remove a document encoding, and
2) require that a successful attack will result in a substantial loss of document presentation quality.

In short, the "cost" of theft should exceed the cost of obtaining a document legitimately. In practice, however, we can realize the above objectives only in part. But establishing any barrier to unauthorized copying and dissemination provides a greater level of document security than publishers now enjoy.

We next describe some illustrative techniques to defeat our document marking methods. We comment on their efficacy and ease of implementation, though we acknowledge that there is a lack of accepted measures to gauge the degree of difficulty necessary for each attack. We also discuss approaches to detect the presence of document coding, though it is generally not necessary to detect the presence of an encoding to defeat it.

A. Defeating the Line-Shift Coding Method

Though line shifts are difficult for the casual reader to discern, they may be found relatively easily by manual or automatic measurement of the number of pixels between text baselines. An attacker can invoke a pixel magnifying glass (a common computer graphics tool) and manually count pixels between adjacent text lines on a page. If adjacent text lines (within a paragraph) are nonuniformly spaced, then the attacker can surmise that encoding has been performed, and precisely measure the displacement introduced by the encoding.

Certain pattern recognition tools, such as the horizontal projection profile, can be used to determine text line spacing automatically. The projection profile is a summation of the "on" pixels along each row in an image; each text line has a corresponding peak in the profile. Therefore, the spacing between adjacent peaks ideally gives text line spacing. However, complications can arise in measuring spacing automatically. For instance, if a document is skewed (line orientations not exactly on the horizontal), then the accuracy of spacings measured by the projection profile will decrease. If a document has multiple columns, each column must first be extracted to measure its line spacing.

Line-shift coding can be eliminated by respacing lines either uniformly or randomly. This requires either manual or automatic cut-and-paste of lines. This process is more difficult to automate if the document contains figures or multiple columns, or is skewed. Respacing lines randomly runs the risk of further decreasing (increasing) line spaces that are already short (long), possibly enabling a casual reader to notice that the document has been tampered.

If a document marked with line-shifting is distributed in paper form, it is particularly challenging to remove the encoding. Of course an attacker can return each page to the electronic domain by scanning, use the above methods, then reprint the document. Removing the encoding from a paper document which itself is a photocopy is even more challenging. Image defects such as component blurring, salt-and-pepper noise, and nonlinear translation within a page, all can potentially combine to disrupt an automated attack.

B. Defeating the Word-Shift Coding Method

The presence of word spacing encoding can be detected in either of two ways. One way is to know or ascertain the spacing algorithm used by the formatter for text justification. Actual spaces between words could then be measured and compared to the formatter's expected spacing. Spacing differences resulting from this comparison would reveal the location and size of text displacements. The second way is to take two or more distinctly encoded, uncorrupted documents and perform a page by page pixel-wise difference operation on the corresponding page images. Such a comparison would quickly indicate the presence of word shifts, and the size of the word displacement.

An attacker can eliminate the encoding by respacing shifted words back to the original spacing produced under the formatter's rule. An alternative attack is merely to apply random horizontal shifts to all words in the document not found at column edges. Word shifting can be done manually using cut-and-paste graphics tools, though producing a document without severe presentation degradation would be time-consuming and painstaking. To perform word-shifting automatically, each text baseline must first be found (perhaps as described in Section IV-A), then individual words segmented, and their spacing changed along the baselines. Words can be segmented by comparing intra-word character spacing to inter-word spacing. However, current segmentation methods are prone to errors introduced by font changes, font size changes, symbols, equations, etc. These complications would likely require manual inspection and intervention for repair, again a time-consuming and painstaking process.

C. Defeating the Feature Coding Method

A document to be feature coded would first be subject to feature randomization prior to encoding. That is, character endline lengths would be randomly lengthened or shortened, then altered again to embed a specific codeword. Using this approach, the location of encoded features cannot be ascertained from inspection of a single document, because the original endline lengths are unknown. Of course, the encoded feature locations can be ascertained by obtaining two or more distinctly encoded, uncorrupted documents, and performing a page by page pixel-wise difference operation on the corresponding page images.

It is interesting to note that a purely random adjustment of endline lengths is not a particularly strong attack on this coding technique. Yet feature encoding can be defeated by using pixel processing tools to adjust each endline length to a fixed value (e.g., the maximum or minimum of the range of lengths observed in multiple, distinctly marked copies of a document). This would obviously be painstaking to do manually, particularly since the number of feature changes introduced in a document can be large (e.g., 1 feature change per word). This attack can be performed automatically, however it can be made more challenging by varying the particular feature to be encoded.

D. Defeating Generic Coding Methods

The attacks we introduced in the previous subsections each work against a specific document marking method. These targeted attacks not only remove the encoding, but can potentially do so with minimal loss of document presentation quality. A second class of attacks exist which can be used to defeat a large class of coding methods, including those introduced in this paper. In general, these broad attacks result in relatively higher loss of presentation quality.

Attacks based on Optical Character Recognition (OCR) are an example. These essentially create a new document by extracting information from the original document images. This process typically requires correctly identifying the locations of both text and image components within a document image (i.e., "zoning"), recognizing characters in textual regions, extracting embedded images, and assembling a new document. In most cases, even a casual observer will recognize that the resulting document differs from the original image in appearance. It is also well known that OCR technology, though widely available and inexpensive, does not always recognize characters correctly. In addition, the current technology used to reconstruct a document format is also imperfect. Hence, some degree of manual intervention may be required of the attacker.

An inherent difficulty in developing a tool for repeated automated attacks is that the tool must work correctly with a rather diverse range of document types. Documents can and do support type in many fonts, arbitrary text layout, encapsulated greyscale or color images, illustrations mixed with text, backgrounds of varying intensities, in-line or displayed mathematics, and an assortment of irregularities that interfere with automated processing.

Even the most powerful automated attacks can be somewhat frustrated by introducing *impediments* within marked documents designed primarily to increase their resistance to attack. For example, word shifting might be used in addition to feature coding, simply to frustrate an attacker's attempts to align corresponding pixels between two distinctly encoded page images. Introducing enough of these impediments to automated attacks might ensure that some manual intervention is required by the attacker.

A technically unsophisticated user of an automated attack tool is ultimately left to consider whether the tool successfully removed all of the markings that might have been included in a document. Presumably the document distributor is able to change document marking techniques occasionally, and can become aware of and exploit the limitations of automated tools. Finally, note that possession of a document with markings either altered or removed indicates that the document is unauthorized. For many users this is sufficient reason to obtain the document from the legitimate source.

V. Summary

Making and disseminating unauthorized copies of documents can be discouraged if each of the original copies is unique, and can be associated with a particular recipient. Several techniques for making text documents unique have been described. One of these techniques, based on text-line shifting, has been described in more detail. A set of experiments was conducted to demonstrate that small variations in line spacing indiscernible to a casual reader can be recovered from a paper copy of the document, even after being copied several times.

In our experiments, the position of the odd numbered lines within each paragraph remained the same while the even numbered lines were moved up or down by a small amount. By selecting different line shifts, information was encoded into the document. To retrieve the information from a paper copy, the document was electronically scanned and analyzed. Two detection methods were considered, one based on the location of the bottom of the characters on each line, and the other based on the center of mass of each line. The advantage of using baselines is that they are equally spaced before encoding and the information can be retrieved without reference to a template. The centers of mass of the lines are not equally spaced, however, this technique has been found to be more resistant to the types of distortion encountered in the printing and copying process.

The differential encoding mechanism was selected because the types of distortion that have been encountered have canceled out when differences between adjacent lines are considered. In the experiments, the lines in the document were moved up or down by as little as 1/300 inch, the document was copied as many as ten times, then the document was scanned into a computer and decoded. For the set of experiments that has been conducted, the centroid decoding mechanism yielded an immeasurably small error rate. Though experiments are ongoing, we believe that sufficient data has been obtained to be convinced that we can identify an intended recipient from a copy of an encoded document as long as it remains legible.

References

[1] M. Lesk, "Full text retrieval with graphics," *Bridging the Communications Gap, AGARD-CP-487*, AGARD, Neuilly sur Seine, France, 1990, pp. 5/1-13.

[2] C. A. Lynch, "Information retrieval as a network application," *Library Hi Tech*, vol. 32, no. 4, pp. 57–72, 1990.

[3] R. Basch, "Books online: Visions, plans, and perspectives for electronic text," *ONLINE*, pp. 13–23, July 1991.

[4] W. Y. Arms *et al.*, "The Mercury Electronic Library and Information System II, The First Three Years," Carnegie Mellon University, Mercury Technical Report Series (6), 1992.

[5] J. H. Saltzer, "Technology, networks, and the library of the year 2000," *Lecture Notes in Computer Science*, 1992 pp. 51–67.

[6] E. A. Fox and L. F. Lunin, "Perspectives on digital libraries," Special issue of *J. Amer. Soc. for Inform. Sci.*, vol. 44, no. 8, Sept. 1993.

[7] M. Hoffman, L. O'Gorman, G. A. Story, and J. Q. Arnold, "The RightPages Service: An image-based electronic library," *J. Amer. Soc. for Inform. Sci.*, vol. 44, no. 8, pp. 446-452, Sept. 1993.

[8] G. A. Story, L. O'Gorman, D. Fox, L. Schaper, and H. V. Jagadish, "The RightPages image-based electronic library for alerting and browsing," *IEEE Computer*, pp. 17–26, Sept. 1992.

[9] L. O'Gorman, "Image and document processing techniques for the RightPages Electronic Library System," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, The Hague, The Netherlands, Sept. 1992, pp. 260–263.

[10] S. Borman, "Advances in electronic publishing herald changes for scientists," *Chemical and Engineering News*, pp. 10–24, June 14, 1993.

[11] J. R. Garrett, "Text to screen revisited: Copyright in the electronic age," *ONLINE*, pp. 22–24, Mar. 1991.

[12] M. Vizard, "Electronic publishing moves ahead," *Computerworld*, p. 37, Mar. 1993.

[13] N. R. Wagner, "Fingerprinting," in *Proc. 1983 Symp. Security and Privacy*, IEEE Computer Society, Apr. 1983, pp. 18–22.

[14] A. K. Choudhury, S. Paul, H. Schulzrinne, and N. F. Maxemchuk, "Copyright protection for electronic publishing over computer networks," *IEEE Network Mag.*, vol. 9, no. 3, pp. 12–21, May/June 1995.

[15] N. F. Maxemchuk, "Electronic document distribution," *AT&T Tech. J.*, vol. 73, no. 5, pp. 73–80, Sept. 1994.

[16] J. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman, "Hiding information in document images," in *Proc. 1995 Conf. Inform. Sci. and Syst.*, Johns Hopkins University, Mar. 1995.

[17] L. O'Gorman, "The document spectrum for structural page layout analysis,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, Nov. 1993.

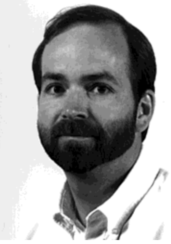
[18] L. O'Gorman and R. Kasturi, "Document image analysis," *IEEE Computer Society Tutorial Text Series*, California, 1994, ch. 4.

[19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992, pp. 152, 301.

[20] H. S. Baird, "Document image defect models," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. Berlin: Springer-Verlag, 1992, pp. 546–556.

[21] L. B. Schein, *Electrophotography and Development Physics*, 2nd ed. Berlin: Springer-Verlag, 1992.

[22] S. H. Low, A. M. Lapone, and N. F. Maxemchuk, "Document marking and identification techniques and their comparison," submitted for publication, 1995.



Jack T. Brassil (M'82–SM'94) received the B.S. degree from the Polytechnic Institute of New York in 1981, the M.Eng. degree from Cornell University in 1982, and the Ph.D. degree from the University of California, San Diego, in 1991, all in electrical engineering.

He has been with AT&T Bell Laboratories since 1981. He is currently a Member of Technical Staff in the Distributed Systems Research Department in Murray Hill, NJ. His current research interests are in broadband networks, intellectual property protection, digital libraries, network applications, and distributed algorithms.

Steven Low (S'88–M'92) received the B.S. degree from Cornell University in 1987, the M.S. and Ph.D. degrees from UC Berkeley in 1989 and 1992, all in electrical engineering.

Since 1992, he has been with AT&T Bell Laboratories. His research interests are in the control and optimization of communication networks, and in the privacy, security and copyright protection issues in the use of networks.

Nicholas F. Maxemchuk (F'89) received the B.S.E.E. degree from the City College of New York, and the M.S.E.E. and Ph.D. degrees from the University of Pennsylvania.

He is currently the Head of the Distributed Systems Research Department at AT&T Bell Laboratories. He has been on the adjunct faculties of Columbia University and the University of Pennsylvania. He has been on the advisory panels for the United Nations, the National Science Foundation, the Rome Air Development Center, the Information Technology Research Center, and the Telecommunications Research Institute of Ontario.

Dr. Maxemchuk has served as an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and is currently a Senior Editor for JSAC and on the Steering Committee for the IEEE/ACM TRANSACTIONS ON NETWORKING. He was awarded the IEEE's 1985 and 1987 Leonard G. Abraham Prize Paper Award.

Lawrence O'Gorman (M'78–SM'93) received the B.A.Sc. degree from the University of Ottawa, Ontario, in 1978, the M.S. degree from the University of Washington, Seattle, in 1980, and the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1983, all in electrical engineering.

From 1980 to 1981 he was with Computing Devices Company in Ottawa, where he worked on digital signal processing and filter design. Since 1984, he has been at AT&T Bell Laboratories, Murray Hill, NJ, in the Information Systems Research Laboratory, where he is a Distinguished Member of Technical Staff. His research interests include document image analysis, pattern recognition, and digital libraries. He has lately been involved in the RightPages electronic library project and aspects of networked document security. He is also co-author of an IEEE tutorial text with R. Kasturi, *Document Image Analysis*.