RC23832 (W0512-071) December 15, 2005
Computer Science

IBM Research Report

Detection and Location of Very Small Print Defects in Real Time for High-Speed Digital Printing

Gordon W. Braudaway
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Detection and location of very small print defects in real time for high-speed digital printing

Gordon W. Braudaway
International Business Machines Corporation, Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

This paper addresses the variations and adaptations of processes needed for verifying print quality in real time for high-speed digital printers. The term high-speed is here defined to be more than 750 printed pages per minute. The processes used for print verification are based on very high speed image processing technology applied to scanned images of the printed pages. Print verification is done by scanning the printed copies on the obverse and reverse web sides thereby forming two streams of digitized images. Digitized images from each scanned image stream are then spatially aligned page by page, line by line and pixel by pixel with digitized images from a corresponding source streams. The accuracy of alignment needed to compensate for the poor dimensional stability of paper creates a need for considerable sophistication in image alignment. The source and aligned scanned images are compared to find pixel sequences that are different. The intent is to identify all differences, either missing-black or unexpected-black, that are larger than a 0.01 inch (0.25 mm) square. These differences may represent unacceptable defects in the printed copies.

Keywords: printing, print verification, image processing

1.0 INTRODUCTION

The variations and adaptations of processes needed for verifying print quality in real time for high-speed digital printers are the subject of this paper. The term high-speed is here defined to be more than 750 printed pages per minute. The processes used for print verification are based on very high speed image processing technology applied to scanned images of the printed pages as they are printed.

A typical high-speed printer with an added two-sided web scanner are shown in Figure 1.

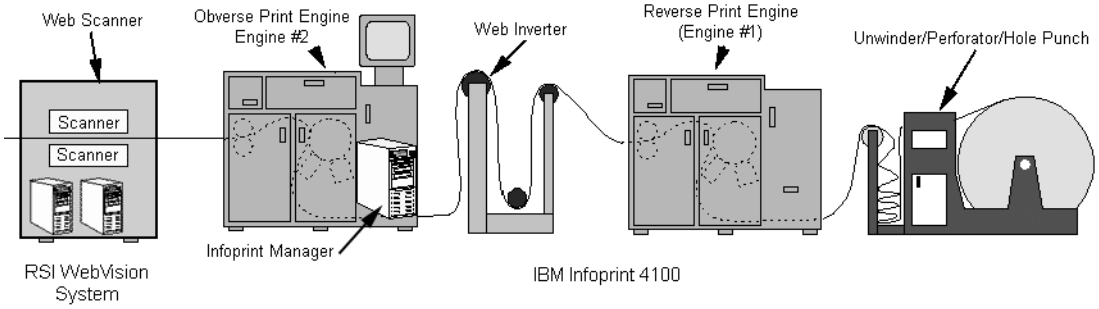

Figure 1. A typical high-speed web-fed print-train, augmented with a two-sided web scanning unit.

Referring to Figure 1, the units, although physically separate, are collectively referred to as a printer or a *print-train*. The units of the print-train are threaded with a common continuous sheet of paper called a *web*. One of the Print Engines is designated as the *Master* in terms of establishing the speed of the web, and all other units in the train independently synchronize their speeds with that of the Master by sensing tension in the web.

The web is presented as roll of paper 18½ inches wide and 40,000 to 50,000 feet in length, depending on paper thickness. The paper roll is mounted into a servomotor controlled Unwinder/Perforator/Hole Punch unit that unwinds and supplies the web to the print train at a speed determined by a Master. The continuous web is perforated into 11½ inch *web-segments*, and guide holes are punched into both edges of the web. Two 8½ by 11 inch page *impressions*, each defined by an image, are printed on the reverse side of each web segment by the Reverse Print Engine (Print Engine 1).

The web is physically inverted by a Web Inverter and passes into the Obverse Print Engine (Print Engine 2) where two additional page impressions are printed on the obverse side of the web. The four impressions printed on each web segment are logically oriented and sequenced by the IBM Infoprint Manager® before being sent to the Obverse and Reverse Print Engines so pages in a final multi-page document will be in the correct order.

The web then passes through the Web Scanner[1] where both the obverse and reverse sides of the web are scanned to produce two streams of digitized scanned images, one stream from the top and the second from the bottom of the web. The web generally continues on into an Accumulator unit followed by either a Separator/Stacker or a Fan Folder unit (not shown).

Although everything that follows applies to either single page or multi-page documents, the multi-page document flow will be used for exposition.

A digitized file containing the pages to be printed, in a format such as PostScript or Portable Document Format (PDF), is submitted to the IBM Infoprint Manager. Each individual document page is transformed into a rasterized image format ready for printing. The Infoprint Manager organizes and places the rasterized images into one of two streams of images, the first stream for the Reverse Print Engine and the second for the Obverse Print Engine. For a document that is to be printed and verified, each printer controller adds various job, alignment, and image processing cues to the rasterized images, and these augmented images are called *source images* (see Figure 2 for "two-up" source image layout). The source images are then printed by the deposition and fusing of dots of black toner onto the paper web. For the purposes of print verification, each source image is presumed to be a correct and unblemished copy of what is intended to be printed. Defects in the original page and in its rasterization must be found by other means, usually by careful visual inspection of a proof copy.

Print verification is done using digitized images from each scanned image stream. The scanned images are spatially aligned page by page, line by line and pixel by pixel with digitized images from a corresponding source stream. The source and aligned scanned images are compared to find pixel sequences that are different. These differences may represent defects in the printed copies or, at the very least, cosmetic differences that may or may not be tolerable.

Some document printing applications require an extremely low likelihood of significant printing defects. Significant defects are defects that could cause misinterpretation of a single printed character by a human reader. A significant defect is defined to be one of two types: 1) either a small square area having black pixel values in the source image that are not found in corresponding pixel values of the scanned image (missing black), or 2) toner or other marking detected in pixel values of the scanned image that is not present in the pixel values of a corresponding source image (unexpected black). Further, to be a significant defect, the unexpected small square of either type must be in the near proximity of a character of text that is intended to be printed. An example small square is of the order of 0.013 inches (0.34 mm) on a side, and an example near proximity is 0.04 inches (1 mm) in any direction. Either of the two types of significant defects constitutes a printing error, and the page on which it is detected and its position on that page are recorded in an error log of defective pages.

A second category of printing defects, called cosmetic defects, are defects that would probably not cause misinterpretation of a single printed character but may still be objectionable in the printed copy. Cosmetic defects include stains on the paper, unintended printed streaks or splotches, wrinkled paper and the like. Cosmetic defects, based on their size, location, and frequency of occurrence, may constitute printing errors. If judged so, the pages on which they are detected along with their position on those pages are also recorded in an error log of defective pages.

Substantial problems become evident when attempting to adapt previously known methods of print quality verification to high speed digital printing. First, and most important, at dimensions approaching 0.01 inch (0.25 mm) the accuracy of alignment required to compensate for the poor dimensional stability of paper creates a need for additional sophistication in the alignment of scanned images with source images. Second, in comparing the scanned image to its corresponding source image, it is necessary to compensate for "dot-gain," that is, for the fact that the size of a single pixel or groups of pixels covers a slightly larger area on paper than they do in the abstract source image. Third, coarse positional alignment of common features found within two images requires selection of a source image corresponding to a scanned image to

be accomplished by automatic means. Fourth, if many copies of the same multi-page document are printed in a sequence as part of a single print job, it is required to automatically match the repeated stream of source images with a particular printed copy of each of those source images, and simultaneously, to identify the page in a particular document in the sequence of documents where each defect lies.

2. "TWO-UP" SOURCE IMAGE LAYOUT

Figure 2 shows the layout of document pages in a source image intended for printing multi-page documents. The documents are to be printed "two-up," that is, two identical documents side-by-side on the web. Note that the left-side document page is inverted relative to the right-side document page. Each printer controller adds various job, alignment, and image processing cues to the source images. These cues, shown in enlarged detail, are the *correlation-strip*, the *synchronization-strip*, the *counter-strip* and the *top-of-form mark*. Note that a synchronization-strip and a counter-strip appear in both the left and right sacrificial margins. In fact, all of the cues lie in sacrificial margins that will be trimmed off after the documents are stitched and bound. The uses of these cues will be explained in the sections that follow.

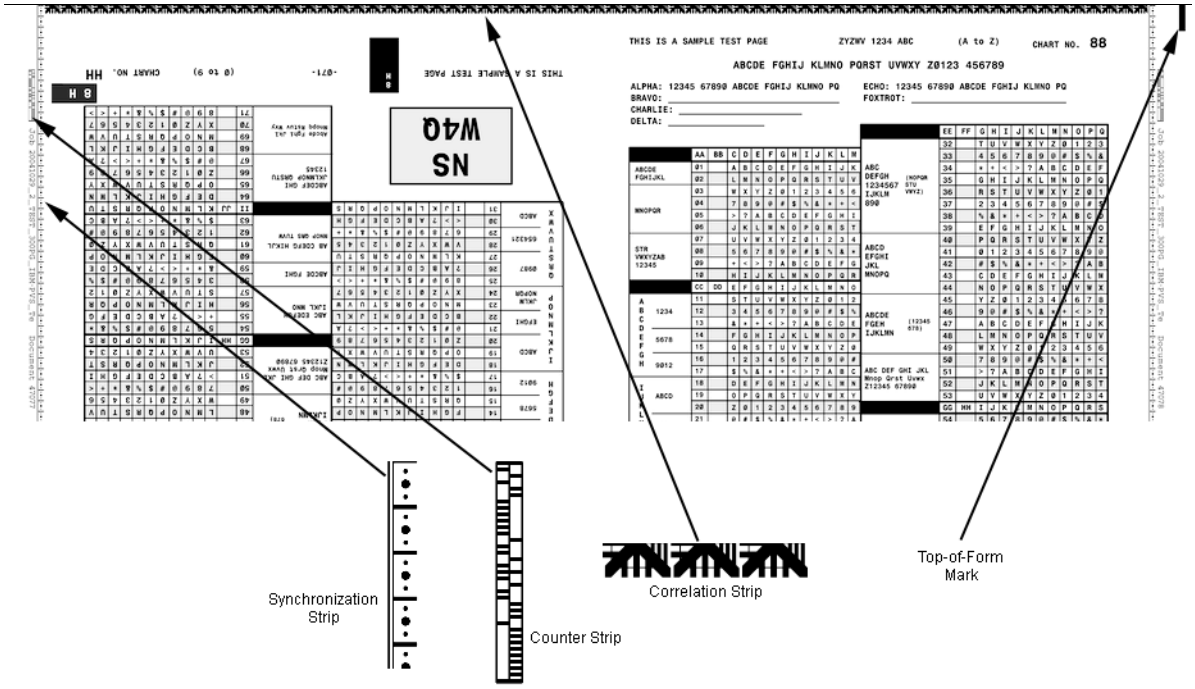

Figure 2. Fragment of a web segment for "two-up" printing showing alignment cues added by the Print Engines.

3.0 DETECTION OF PRINTING DEFECTS

The objective of print verification is to detect all printing defects larger than 0.01 inches (0.25 mm) square. A typical printer, as described above, prints documents at 600 raster lines per inch (23.6 per mm), with each line having 600 pixels per inch. Detection of printing defects can logically be divided into two parts. The first part is to align an image from one of the scanned image streams with a corresponding image from a source image stream. Once accurate alignment of the images is achieved, the second part is to determine the locations of black pixels in the source image that do not exist in the scanned image (missing black) or vice-versa (unexpected black). Of the two parts, image alignment is by far the more computationally demanding. As will be shown, the unpredictable warping of the paper across and along the web is a significant problem. Warping which occurs between the time toner is applied and the time it is thermally fused to the paper is generally several times larger than 0.01 inch defect size being looked for. Thus, the quality of alignment is fundamental to successful verification.

3.1 Alignment of scan and source images

Spatial alignment of digitized images from a scanned stream with those of a corresponding source stream is

accomplished in three phases: a coarse alignment phase, a horizontal fine alignment phase, and a vertical fine alignment phase. The objective of coarse alignment is to locate lines in the scanned stream that correspond to lines in the source stream. In the chosen embodiment, this is facilitated by linear reference markers, called *synchronization-strips*.

Referring to Figure 2, lines of two page impressions lie horizontally across the web, each impression being a page of one of two side-by-side identical document copies being printed. The two impressions and the surrounding alignment cues, collectively called a source image, when printed will be referred to as a *web segment*. The left synchronization-strip, aligned vertically (along the web), is shown in the left trim edge of the web segment. An enlargement of the synchronization-strip shows the basic pattern that is repeated along the edges of the source image. A left *counter strip* is placed adjacent to the left synchronization-strip. The counter strip contains two rudimentary 16-bit binary counters. The filled square in the counter pattern lying rightmost represents a zero for that particular bit in the counter, and a filled square in the counter pattern lying leftmost represents a one. With the bits read from bottom to top, the top 16-bit counter contains the document copy number, and the bottom counter contains the page number within that document.

A like synchronization-strip and counter-strip are placed in the right trim edge of the web segment as well, the only differences being the numeric value of the document copy number (which differs by one), and their placement relative to each other (the counter is always placed nearer to the web edge).

The synchronization-strips and counter-strips embedded into the source stream images will, when printed, exist in printed web segments, and when the web segments are scanned by the Web Scanner, will also exist in the scanned stream images. Coarse alignment of corresponding images of the source stream with the scanned stream relies on matching the patterns in the synchronization-strips to within a fraction of a line width and a fraction of a pixel position.

Images in the scanned stream have fewer lines per inch than do images in the source stream. The digitized images in the source stream have 600 lines and pixels per inch while images in the scanned stream have approximately 280 lines and pixels per inch. Thus, to be able to match images of the two streams line by line, additional lines of the second image must be created between the available scanned lines using interpolation methods. Conversely, images of the source stream could be reduced to have 280 lines per inch using any of many image reduction methods. The computationally more efficient method, however, is to enlarge images in the scanned stream by interpolation and then convert the enlarged number of 8-bit pixel values to 1-bit pixel values.

3.1.1. Synchronization-strip tracking

To facilitate interpolation between the pixels of images in the scanned stream, a complementary pair of affine transforms are created for each half-inch (300 line) strip of the source image. The values required to establish each affine transform are determined by independently *tracking* the left and right synchronization-strips. The term *tracking* as used here refers to the numeric evaluation of horizontal and vertical coordinates of *specifically identifiable features* in the synchronization-strips in the scanned images, and relating those values to known numeric values of horizontal and vertical coordinates of the same features embedded into the corresponding source image. The counter-strips are also read by the tracker and are used to establish gross correspondence between scanned and source images from the two image streams; that is, to locate the required corresponding source image in its stream for each scanned image. To add robustness to identification of a particular scanned image, the tracker takes advantage of the redundant nature of data within the counter-strips themselves. Each counter-strip contains both a document and page count and the ones-complement of each of the counts, and the strips are essentially redundant left-side to right side, differing only by a count of one in the document value.

An example embodiment of a tracker uses localized application of a two-dimensional cross-correlation function, repeatedly and sequentially applied, to determine the values of horizontal and vertical coordinates of the features in the digitized scanned images, with linear interpolation for other coordinate values between those discrete applications. Details of the tracker are beyond the scope of this paper, but those familiar with signal and image processing will recognize that the tracking method can be embodied in a number of functionally equivalent ways that produce the required results.

Because of the difference in pixel resolution, additional pixels in lines of the scanned stream image must also be created by interpolation to be able to closely match individual pixel positions within corresponding pairs of images in the two streams. Therefore, the interpolation method needed is a two-dimensional method capable of interpolating between available lines to produce intermediate lines of pixels and interpolating between available pixels in those lines to produce intermediate pixel values.

3.1.2 Coarse alignment

Four *coordinate-pairs* from each image in the source stream and four *coordinate-pairs* from a corresponding image in the scanned stream are used for coarse alignment of a horizontal strip of each web segment. The height of the strip is one-half inch, or 300 lines of the source image. Figures 3 and 4 illustrate the source image and the scanned image. The locations of corresponding *specifically identifiable features* of the synchronization-strips, as determined by the tracker, are noted in both images.

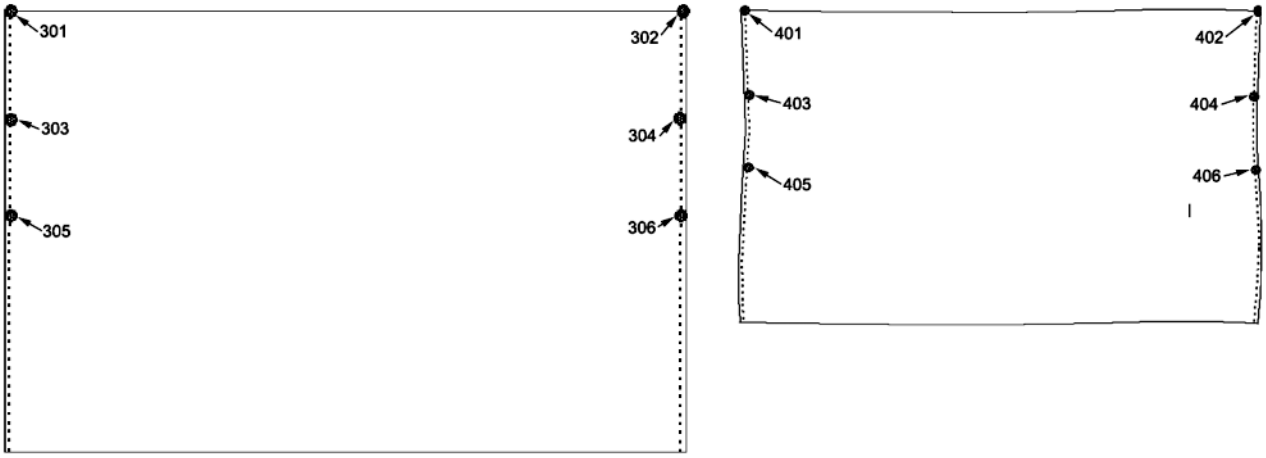

Figure 3: Source Image (left) and Figure 4: Scanned Image (right) showing positions of corresponding features, as determined by the Tracker.

The first four reference points from the source image are numbered (301) through (304) and the first four from the corresponding scanned image (401) through (404). If (u, v) are the horizontal and vertical coordinates pair of pixels in the scanned image, and (x,y) are the horizontal and vertical coordinates pair of pixels in the source image, the coordinates pairs can be related by an affine transformation. In matrix-equation form, the affine transformation is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} {}_n\mathrm{U}_x & {}_n\mathrm{U}_y \\ {}_n\mathrm{V}_x & {}_n\mathrm{V}_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} {}_n\mathrm{U}_c \\ {}_n\mathrm{V}_c \end{bmatrix} \tag{1}$$

where the coefficient $\{{}_n\mathrm{U}_x, {}_n\mathrm{U}_y, {}_n\mathrm{U}_c, {}_n\mathrm{V}_x, {}_n\mathrm{V}_y, {}_n\mathrm{V}_c\}$ are constants. (Note, the pre-subscript, n, refers to a particular instance of the coefficients, and each instance is valid for interpolation only in a triangular area of pixels bounded by three specific reference points, as will be shown next). To determine the six coefficients of the first instance, Equation (1) can be expanded and rewritten in terms of three coordinate-pairs from the source and scanned images as Equation (2), where the subscripts 1, 2, and 3 for (x,y) are connected to coordinate pairs of reference points (301), (302), and (303), and for (u,v) are connected to coordinate pairs for reference points (401), (402), and (403).

$$\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} {}_1\mathrm{U}_x \\ {}_1\mathrm{U}_y \\ {}_1\mathrm{U}_c \\ {}_1\mathrm{V}_x \\ {}_1\mathrm{V}_v \\ {}_1\mathrm{V}_c \end{bmatrix} = \boldsymbol{X}_1 \begin{bmatrix} {}_1\mathrm{U}_x \\ {}_1\mathrm{U}_y \\ {}_1\mathrm{U}_c \\ {}_1\mathrm{V}_x \\ {}_1\mathrm{V}_v \\ {}_1\mathrm{V}_c \end{bmatrix} \tag{2}$$

Equation (2) can be solved to give a first-instance of the coefficients $\{{}_1U_x, {}_1U_y, {}_1U_c, {}_1V_x, {}_1V_y, {}_1V_c\}$, as shown in Equation (3a) (the superscript -1 is taken to mean the inverse of that matrix).

$$\begin{bmatrix} {}_1U_x \\ {}_1U_y \\ {}_1U_c \\ {}_1V_x \\ {}_1V_v \\ {}_1V_c \end{bmatrix} = X_1^{-1} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} \tag{3a}$$

A second instance of the coefficients, $\{{}_2U_x, {}_2U_y, {}_2U_c, {}_2V_x, {}_2V_y, {}_2V_c\}$, is determined using a similar Equation (3b).

$$\begin{bmatrix} {}_2U_x \\ {}_2U_y \\ {}_2U_c \\ {}_2V_x \\ {}_2V_v \\ {}_2V_c \end{bmatrix} = X_2^{-1} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} \tag{3b}$$

The positions of the scanned pixels contained in the trihedrons having apexes (401),(402),(403) and (402),(403),(404) relative to the positions of the source image pixels contained in the trihedrons having apexes (301),(302),(303) and (302),(303),(304) are determined using Equation (1), with application the first-instance or second-instance of the coefficients, $\{{}_1U_x, {}_1U_y, {}_1U_c, {}_1V_x, {}_1V_y, {}_1V_c\}$ or $\{{}_2U_x, {}_2U_y, {}_2U_c, {}_2V_x, {}_2V_y, {}_2V_c\}$, that is appropriate for the specific location of (x, y). Applied in this manner, the affine transform of Equation (1) determines the two-dimensional coarsely aligned positions of the interpolated pixels, called *points of interest*. Note that points of interest on the (402),(403) diagonal are the same whether determined using the first-instance or second-instance of the coefficients.

The interpolated pixel brightness values at the *points of interest* are determine by linear area interpolation in the two-dimensional $u : v$ plane. A small square *sub-area* of the $u : v$ plane (which is the plane of the scanned image) is isolated using the integer parts of the point of interest. For example, if the interpolated coordinates (u,v) are (1303.297,457.338), the point of interest is surrounded by the four pixel locations (1303,457}, (1304,457), (1303,458) and (1304,458), and those four pixel locations, called *vertices*, define the square sub-area of interest. The fractional parts the coordinates (u,v) are called the *residuals*, (u_R, v_R), and are numerically (0.297,0.338).

Linear interpolation in the sub-area begins by dividing the unit square into two trihedrons along a diagonal. Of the two choices available for dividing the square, the upper-left to lower-right diagonal connecting vertices (1303,457) and (1304,458) is chosen arbitrarily. Then the trihedron in which the point of interest lies is determined. The criterion for selection is obvious: if $u_R \geq v_R$, the upper-right trihedron is chosen because it will completely enclose the point of interest; otherwise, if $u_R < v_R$, the lower-left trihedron is chosen. If the point of interest lies on the diagonal, $u_R = v_R$, either trihedron will do, since both will produce the same numeric result.

Interpolation within the trihedron can be defined in terms of the *natural coordinates* of the point of interest. All natural coordinates will be the in domain (0,1) for every interpolated point of interest, and at least one will not be in the domain for every extrapolated point of interest. Interpolation within a trihedron can be defined in terms of the natural coordinates, a_i, of a *point of interest* as:

$$f(u_R, v_R) = a_A f(u_A, v_A) + a_B f(u_B, v_B) + a_C f(u_C, v_C), \quad \text{where} \sum_{k=A,B,C} a_i = 1 \tag{4}$$

The rectangular coordinates of the vertices are denoted as (u_k, v_k). The dependent variables at the vertices, which in the

example are the pixel brightness values, are denoted as the functions $f(u_k, v_k)$. A further side condition is that the natural coordinates must sum to one. Expressing this relationship in matrix form:

$$[f(u_R, v_R)] = \begin{bmatrix} f(u_A, v_A) & f(u_B, v_B) & f(u_C, v_C) \end{bmatrix} \begin{bmatrix} a_A \\ a_B \\ a_B \end{bmatrix} \tag{5}$$

Based on the same natural coordinates, the coordinates of the point of interest can be written in terms of the rectangular coordinates of the vertices, as

$$\begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix} = \begin{bmatrix} u_A & u_B & u_C \\ v_A & v_B & v_C \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_A \\ a_B \\ a_B \end{bmatrix} \tag{6}$$

Thus, the interpolation equation can be written as

$$[f(u_R, v_R)] = \begin{bmatrix} f(u_A, v_A) & f(u_B, v_B) & f(u_C, v_C) \end{bmatrix} \begin{bmatrix} u_A & u_B & u_C \\ v_A & v_B & v_C \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix} \tag{7}$$

For the example interpolation, the independent values u_R and v_R are 0.297 and 0.338, respectively. Because u_R is less than v_R, the lower-left trihedron having vertices (1,1), (0,1) and (0,0) is chosen. The natural coordinates are computed as:

$$\begin{bmatrix} a_A \\ a_B \\ a_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.297 \\ 0.338 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0.297 \\ 0.338 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.297 \\ 0.041 \\ 0.662 \end{bmatrix} \tag{8}$$

All of the natural coordinates are positive and in the domain (0, 1), and their sum is 1.0, as expected.

It is worth noting in passing that since the sub-area of interest will always be a unit square, the coordinates of its vertices will always be zeros or ones. Because of that, the inverse of the matrix of coordinates can have only values of 1, 0 , and -1 for its elements. Thus, the natural coordinates can always be computed without multiplication, using only addition and subtraction. This is an important consideration, because the two-dimensional interpolation of pixel brightness values of the scanned image is the most computationally demanding part of the entire image alignment process.

Construction of a coarsely aligned replacement for a strip of the scanned image can now be completed. The entire coarsely aligned replacement of the scanned image, including this first strip and all subsequent strips, will be referred to as the *initial replacement image*. For every pixel location in or on the boundary of the rectangle defined by the four coordinate-pairs from the source image (301),(302),(303),(304), a *point of interest* is computed using Equation (1) applying the appropriate first or second instance of the coefficients, $\{{}_1U_x, {}_1U_y, {}_1U_c, {}_1V_x, {}_1V_y, {}_1V_c\}$ or $\{{}_2U_x, {}_2U_y, {}_2U_c, {}_2V_x, {}_2V_y, {}_2V_c\}$. At each point of interest, a pixel brightness value is interpolated using Equation (7). The interpolated pixel brightness value is placed into an array of pixel brightness values that have a one-to-one positional correspondence with the source image pixel location used to evaluate the point of interest. In this manner a strip of the *initial replacement image* is constructed that has the same number of lines and pixels per line as the source image rectangle, and it is placed in the same location in the initial replacement image as the strip bounded by the four coordinate-pairs (301),(302),(303),(304) has in the source image. To simplify further image processing, the interpolated pixel brightness values are *binarized*, that is, they are converted to one of two values and represented by a single binary

bit. The criterion used for binarization is simple thresholding; if the pixel brightness value is less than a threshold value it is set to 1; otherwise, it is set to 0. An example threshold value is 50% of the maximum brightness value. A binarized pixel value is a special case of an interpolated scanned pixel value represented by a single bit; in this embodiment, black pixel values are set to one and white pixel values to zero.

Construction of the remaining strips of the initial replacement image is continued using the next groups of four coordinate-pairs from the source image (303),(304),(305),(306) and the group of four corresponding coordinate-pairs from the scanned image (403),(404),(405),(406) to produce the third and fourth instances of the coefficients, $\{{}_3U_x, {}_3U_y, {}_3U_c, {}_3V_x, {}_3V_y, {}_3V_c\}$ or $\{{}_4U_x, {}_4U_y, {}_4U_c, {}_4V_x, {}_4V_y, {}_4V_c\}$. The strip of the source image bounded by the four coordinate-pairs (303),(304),(305),(306) is constructed in like manner as above by applying the third and fourth instances of the coefficients. The process of constructing a second strip of the initial replacement image is completed in a like manner as for the first strip. Additional strips are constructed in like manner until all strips of the first web segment are completed. Then the process defined for the first web segment is repeated for every subsequent web segment until the entire document is completed, and then repeated as many times as necessary for all document copies that define a printing job.

3.1.3. Compensating the scanned pixels for sensor variations

Pixel brightness values, denoted as the functions $f(u_i, v_i)$, are not taken directly from a scanned image of the second image stream. Since the value $f(u_R, v_R)$ is computed by interpolation and is a weighted average of values of its three surrounding neighbors $f(u_A, v_A)$ $f(u_B, v_B)$ $f(u_C, v_C)$, and since each pixel value is digitized by a separate sensor in one of two physically separate linear arrays [top or bottom of the web] in the Web Scanner, each of the sensing elements needs to be compensated individually in terms of the black and white values it senses. If there are J pixel brightness sensing elements for each line, and B_j and W_j are the measured pixel brightness values when the sensor j $(1 \le j \le J)$ is sensing black (illumination off) and blank paper (illumination on), then, for line i, the pixel brightness value compensation equation is:

$$f(u_j, v_i) = \min[\, \max[\, (f^*(u_j, v_i) - B_j)/(W_j - B_j), 0], 1] \tag{9}$$

where $f^*(u_j, v_i)$, are the uncompensated brightness values measured by the individual sensing elements.

3.1.4 Horizontal fine alignment of images of the source and scanned streams

Further alignment of pixels in the *initial replacement image* relative to corresponding pixels in the *source image* is needed in the horizontal direction [across the web] (and, also, in a lesser degree in the vertical direction [along the web]) because of the unpredictably bad behavior of paper. In the small dimension of a 0.01 inch, paper warps, shrinks and expands based on its moisture content, and its moisture content varies rapidly based on the temperature and humidity of objects it comes in contact with. Horizontal fine alignment of pixels in the *initial replacement image* produces yet another image, called the *intermediate replacement image*.

Horizontal fine alignment of pixels requires slicing the horizontal strips of the initial replacement image into vertical stripes. Fine alignment is achieved by computing the one dimensional [horizontal] cross-correlation function of the pixel values of the *initial replacement image* and the *source image* on a strip-by-strip and a stripe-by-stripe basis. The stripes are chose to be 160 pixels wide and 300 lines high. Thus, each line of the stripe is represented in the computer's memory by five 32-bit words. As a computational convenience, each line of the initial replacement image is padded by 64 pixels of white (two 32-bit words set to zero) on its left end.

Evaluation of the horizontal cross-correlation between the *initial replacement image* and the *source image* is begun with the leftmost stripe of the strip. Binary pixel values are referenced thirty-two at a time, that is, as one 32-bit word. The index m, $(1 \le m \le 300)$ is used to reference the lines of the strip and the index n, $(1 \le n \le 5)$ is used to reference groups of 32-pixel values [32-bit words] in each line of the stripe.

The cross-correlation function is denoted as $C(k)$, where k is the number of pixel positions of horizontal offset in lines of the *initial replacement image* relative to pixel positions in lines of the *source image*. Where k is positive, pixels of the *initial replacement image* are offset to the right of pixels in lines of the *source image*. Because pixels values of both

images are represented by one bit, very significant simplification can be achieved in computing the cross-correlation function. Multiplication of pixel brightness values in the cross-correlation function can be replaced by a bit-wise logical ***exclusive-or***, symbolized herein as ⊛, which operates on thirty-two pixel values at a time. The summation of sub-products in the cross-correlation function is replaced by a sum of the count-of-ones remaining after application of the 32-bit exclusive-or operation.

For an example single stripe, five words in width and with pixels in lines of the *initial replacement image* offset thirty-two pixel positions [one word] relative to pixels in lines of the *source image*, the simplified cross-correlation function value is given by:

$$C(32) = \sum_{n=1}^{5} \sum_{m=1}^{300} \text{count1s}(S(m,n) \circledast R_{[0]}(m,n-1)) \tag{10}$$

where $S(m,n)$ refers to 32-pixel words of the source image line, $R_{[0]}(m,n-1)$ refers to 32-pixel words of the initial replacement image line, and the function "count1s" returns the count of bits in its 32-bit argument that have the value 1. [Note that the index value $n-1$ causes pixels of the initial replacement image lines to be offset thirty-two pixel positions to the right of pixels in the source image lines, which, initially, is redundantly indicated by the zero subscript of R].

Computation of another value in the cross-correlation function of the stripe, for example, $C(30)$, requires that all pixel values in lines of the *initial replacement image* strip are physically shifted two pixel positions to the left. The pixel values of the two vacated pixel positions at the right end of each line are filled with zeros. The procedure is repeated until the pixel values have been shifted left 64 positions, and the corresponding value in the cross-correlation function of the stripe, $C(-32)$, is evaluated. The process, left shifting pixels two pixel positions at a time, leads to 65 evaluations of the cross-correlation function for the single stripe.

The best alignment of the two image stripes occurs at the least value of the cross-correlation function. [Note that a perfect alignment of the two image stripes, black pixels to black pixels and white pixels to white, produced by some shift value k will produce a $C(k)$ having a value of zero.] By taking the offset pixel location k^* of the smallest value of at the cross-correlation function evaluation $C(k^*)$, together referred to a pair, and two additional pairs of values, one on either side of the minimum, a parabola can be fit through the three pairs of values. The pixel position value at the inflection point of the parabola, $k_{\min}$, is a very good estimate of the best alignment between the *initial replacement image* and the *source image* for the particular stripe in the particular strip. It is, however, only an average value of the alignment across the stripe, and it does not include any vertical alignment correction

There are, of course, pathological cases where the horizontal cross-correlation function applied to a stripe will produce an indefinite inflection point. An example that produces an obvious indefinite inflection point is a stripe intended to be blank paper. In that case, regardless of the number of pixel positions a pixel in lines of the *initial replacement image* is displaced relative to a pixel in lines of the *source image*, the count-of-ones will always have a zero sum, and $k_{\min}$ can not be evaluated. A *blank-paper screen* that will detect this particular case uses the logical ***or*** operator to combine all pixel values of every line of the source image stripe. If the logical ***or*** of all pixel values in the stripe has a zero value, then the stripe is intended to be blank paper. For this and other pathological cases that produce an indefinite inflection point, the inflection point from the same stripe in the immediately preceding strip is used instead.

This method of handling indefinite inflection points requires that a carefully constructed source image strip guaranteed to produce no indefinite inflection points is initially printed in each web segment. This is the purpose of the *correlation-strip* added to the top of the source image (see Figure 2). The correlation-strip insures that every stripe in subsequent strips will have a preceding stripe possessing defined values for their inflection points, and if necessary, the inflection points will be passed on from strip to strip to strip.

Values of $k_{\min}$ in every stripe of strip, along with horizontal edge alignments determined by the tracker, are used to

form a *horizontal offset function*, ${}_nH(x)$. Intermediate values of ${}_nH(x)$ lying between two values of $k_{\min}$ are determined by straight line interpolation. Thus, for every source image pixel position, a further correcting horizontal offset of pixel position in the initial replacement image is estimated, and that further correcting horizontal offset, called a *horizontal fine alignment*, is applied in addition to the affine transform; taken together, they form the *points of interest* for the pixel positions of the *intermediate replacement image* from the corresponding *scanned image* of the scanned stream. The two corrections can be made simultaneously using a modified Equation 1, as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} {}_nU_x & {}_nU_y \\ {}_nU_x & {}_nU_y \end{bmatrix} \begin{bmatrix} {}_nH(x) + x \\ y \end{bmatrix} + \begin{bmatrix} {}_nU_c \\ {}_nU_c \end{bmatrix} \tag{11}$$

The re-interpolated values of the *intermediate replacement image* pixel values can then be determined by reapplying Equation (7) using value (u, v) determined from Equation (11), with the appropriate instances of the coefficients $\{{}_nU_x, {}_nU_y, {}_nU_c, {}_nV_x, {}_nV_y, {}_nV_c\}$. Note that values of (x, y) will be pairs of integers, but values of (u, v) will generally not be.

Figure 5 shows the three typical evaluations of the horizontal offset across the web. The evaluations are from the top strip (which included the *correlation-strip* pattern), a middle strip and the bottom strip. Note that warping of the paper of the web shifts printed material from -6 to +10 pixel positions (-0.01 to +.017 inches), a very substantial distance if defects in printing of the order of 0.01 inches are to be found and correctly classified. Warping of this magnitude verifies the need for horizontal fine alignment.

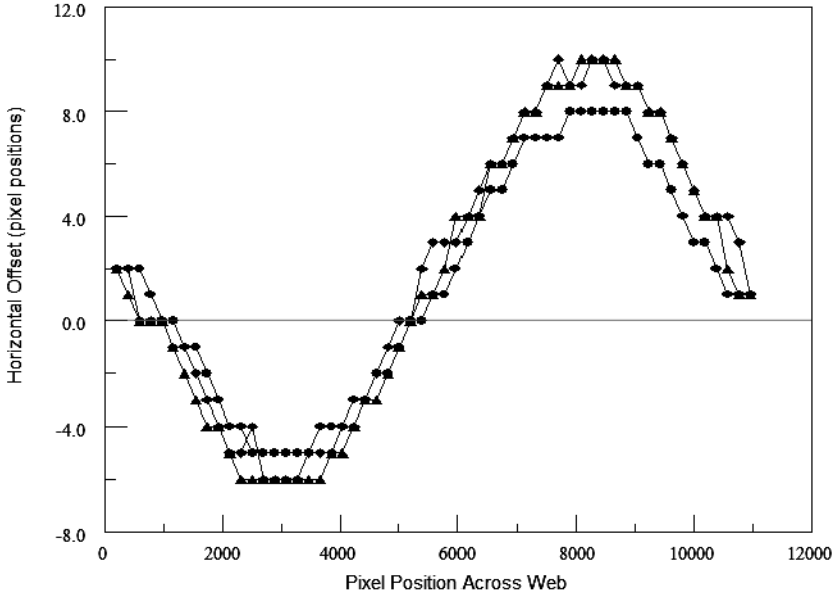

Figure 5: Typical horizontal pixel offset variations at the top, center and bottom of a web segment.

3.1.5 Vertical fine alignment of images of the source and scanned streams

Further fine alignment of pixels in the *intermediate replacement image* relative to corresponding pixels in the *source image* is needed in the vertical direction [along the web]. As before, fine alignment of pixels requires slicing the horizontal strips of the intermediate replacement image into vertical stripes. Vertical fine alignment is facilitated by computing the one dimensional [vertical] cross-correlation function of the pixel locations of the *intermediate replacement image* with those of the *source image* on a strip-by-strip and a stripe-by-stripe basis. The stripes, as before, are chose to be 160 pixels wide and 300 lines high. The image formed by vertical fine alignment is called the *final replacement image*.

Evaluation of the vertical cross-correlation between the *intermediate replacement image* and the *source image* is begun

with the leftmost stripe of the strip. Binary pixel values are again referenced thirty-two at a time, and combined by a bit-wise logical ***exclusive-or***. The summation of sub-products in the cross-correlation function is replaced by a sum of the count-of-ones remaining after application of the 32-bit exclusive-or operation, as before. The principal difference in the computation of the vertical cross-correlation function is that the scanned image lines in each stripe are offset vertically (which is done by line indexing) rather than horizontally (which was done by pixel shifting within a line). The interpolated *vertical offset function* of ${}_nV(x)$, similar in form and function to the *horizontal offset function* ${}_nH(x)$, is derived in a like manner.

A very significant simplification can be applied to construction of a *final replacement image*. A third pixel brightness value interpolation is not warranted. Rather, 32-bit groups of binarized pixels from the *intermediate replacement image* can be offset at a time, either up or down, according to the appropriate truncated value from the *vertical offset function* ${}_nV(x)$, and copied to form lines of the *final replacement image*. Vertical offsets are of the order of $n \pm 2$ pixel positions, substantially smaller than the horizontal offsets.

3.2 Developing comparison masks from the *final replacement image* and the *source image*

In the described embodiment, with binarized *source image* and its corresponding *final replacement images* which are both the same size, direct comparison of pixel values can be done in an efficient manner by employing 32-bit logical operators. To do the comparison, two *masks* are formed from the pixel values of each of the two images. Each mask is itself another binariazed image with size equal to that of the source image. The four masks, specific to each image, will be referred to as the *source dilation-mask*, the *source erosion-mask*, the *replacement dilation-mask* and the *replacement-erosion mask*.

The *source dilation-mask* and the *replacement dilation-mask* are formed by a process called *dilation*. Construction of a *source dilation-mask* is started by setting all pixel values in the mask to zeros, which represent white pixels. The pixel position of each black pixel in the source image forms a pixel-center at the same location in the dilation-mask, and a pixel-array centered at the pixel-center is filled with black pixels in the mask. In the preferred embodiment, a pixel-array with a size designation of "three" is represented by nine black pixels arranged in a 3x3 pixel square. If the dilation size designation is an even integer *P*, the pixel array is represented by P^2black pixels in a *P* by *P* array, and the pixel-center of the array is arbitrarily chosen to be the upper-left pixel closest to the center of the pixel array. The *replacement dilation-mask* is formed in an identical manner. Dilation-masks formed in this manner produce text, lines and halftone patterns that appear to be made of thicker strokes than those of the original image, that is, they appear "dilated" (see Figure 6).

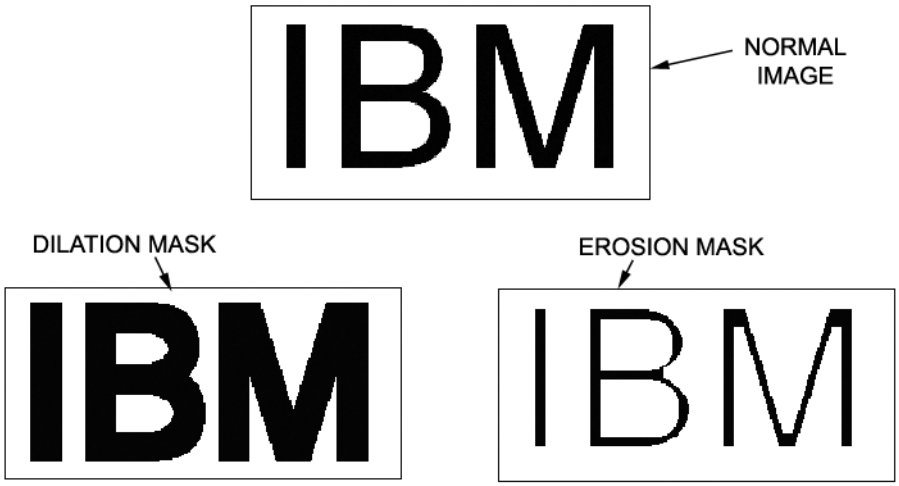

Figure 6. Dilation and erosion masks created from a normal image.

In a similar manner the *source erosion-mask* and the *replacement erosion-mask* are formed by a process called *erosion*. Construction of the *source erosion-mask* is started by setting all pixel values in the *source erosion-mask* mask to ones,

which represent black pixels. The pixel position of each white pixel in the source image forms a pixel-center at the same location in its corresponding source erosion-mask, and a pixel-array centered at the pixel-center is filled with white pixels in the mask. The *replacement erosion-mask* is generated from the *final replacement image* in an identical manner Erosion-masks formed in this manner produce text, lines and halftone patterns that appear made of thinner strokes than those of the original image, that is, they appear "eroded. (again see Figure 6).

3.3 Detection of significant defects in the printed copy

Comparison of pixels in the *final replacement image* with those in the *source image* is done using the four generated masks. Dilation and erosion masks are used to allow tolerance of a small positional uncertainty in the alignment of the *final replacement image* with the *source image* and to compensate for "dot-gain." In all cases, the source image is considered to be an error-free copy of what was intended to be printed. The final replacement image, whose origin is directly traceable to a scan of the printed copy, may not be error free because of defects in the printing process. The binarized pixel values of the *source dilation-mask*, the *source erosion-mask*, the *replacement dilation-mask*, and the *replacement erosion-mask* will be referred to as $S_D(x,y)$, $S_E(x,y)$, $T_D(x,y)$, and $T_E(x,y)$, respectively, and where (x,y) are the indices of the image line,, $_n y$ and pixel location within that line, x. Comparison of the pixel values of specific pairs of these images is done using logical operators.

The preferred embodiment used for detecting unintended black in a single pixel on the printed page is the equation:

$$[S_D(x,y) \oplus T_E(x,y)] \circledast S_D(x,y) = 1 \tag{12}$$

The symbols $\oplus$ and $\circledast$ are the single-bit logical operators, ***or*** and ***exclusive-or***. If the evaluation of Equation (12) is a 1, an unintended black pixel is detected. Since digital computers are capable of performing bit-wise logical operations on 32 pixels with a single instruction, depending on the computer used, significant efficiency of comparison of pixel values is obtained.

The preferred embodiment for detecting the missing black in a single pixel on the printed page is the equation:

$$[S_E(x,y) \oplus T_D(x,y)] \circledast T_D(x,y) = 1 \tag{13}$$

If evaluation of Equation (13) is one, a missing black pixel is detected.

To obtain a lower false detection probability, a small cluster of detected pixels, for example, a 2x2 pixel square area, is used in which a detection must be made in two adjacent pixels in two adjacent lines before a detection is deemed to be significant. This is consistent with the premise that at 600 pixels per inch and 600 lines per inch no meaningful information is conveyed to a human viewer by a single pixel.

4.0 RESULTS AND CONCLUSIONS

The Print Verification System described above has been implemented, and extensive testing has shown it works well. Detection of 8x8 pixel seeded defects have been above 98% for missing black and 95% for unintended black. For larger defects, detection approaches 100%. False detects, that is, the call out of a defect when subsequent visual inspection under magnification finds no defect present, have remained relatively small at about two false defect per 10,000 pages printed.

In fact, the detection process has worked too well at finding cosmetic defects in the paper that are of no practical consequence. Figure 7 shows a cosmetic defect in the paper that is roughly circular and approximately twelve pixels in diameter. To reduce the number of documents discarded because of this kind of defect, a proximity screen has been implemented. The proximity screen has two stages. The first stage estimates the size of the detected defect, and if it is smaller than a specified value (presently 0.05 inch [1.25 mm] square) it is passed to the second stage. The second stage determines its proximity to intended black in the source image, and if it is greater than 0.04 inches distant in every direction from the intended black, it is allowed to pass without a "defect-detected" classification. This has substantially reduced the number of cosmetic defects that are called out.

Figure 7: Scanned Image (left) showing defect in the paper, with Detection Image (right) showing detection of the defect (arrow) as "unexpected-black."

An example of a seeded defect is shown in Figure 8. Black pixels are intentionally imbedded into a source image, which is then compared to a scanned image of an identical page that has no imbedded defect. The largest square area in the seeded defect that does not overlay pixels of the letter "o" is 7x7 pixels. The left-side image of Figure 8 shows the source image with the embedded defect. The right-side image shows detection of that defect by the detection masks. The dark gray area is the *replacement dilation-mask*, the black area is the *source erosion-mask* (with embedded defect) and the light gray area (arrow) are the pixels extending outside the dilation mask. The light gray area contains a 2x2 pixel square, and if the detection screen were set at 2x2 pixels, a "missing-black" defect is reported. If the detection screen were set at 3x3 pixels, no defect detection would be reported.

Figure 8: Source Image (left) with seeded 7x7 pixel square and Detection Image (right).

A proof-of-concept system has been fully developed and is in daily production service. It is capable of detecting defects in one out of every six web segments printed. Adding additional computing power will bring this ratio closer to detecting defects in every web segment.

Although the expository Print Verification System was for printing "two-up" identical multi-page document, the process is easily adapted to printing single page documents, such as financial statements where each page has unique content.

5.0 ACKNOWLEDGMENTS

As with most endeavors of its magnitude, development and embodiment of the Print Verification System was a collaborative effort. Among the several people who made it a success, I specifically acknowledge the very significant contributions of Myron "Flick" Flickner, of the IBM Almaden Research Center, and Rose Visoski and Ronald Parrish, of the IBM Systems & Technology Group. Dr. Flickner designed and implemented the critical Tracker function. Ms. Visoski embedded generation of the alignment cues into the printer controllers. Mr. Parrish designed and implemented the overall synchronization of printers and web scanner with the tracking and verification functions, and coordinated the entire software configuration.

REFERENCES

1. The Web Scanner used is the **Web Vision System (WVS)** manufactured by Lasermax Roll Systems, a division of Stralfors Inc.