High-speed Document Sensing and Misprint Detection in Digital Presses

Guillaume Leseur[a], Nicolas Meunier[a], Georgios Georgiadis[a], Lily Huang[a], Jeffrey DiCarlo[b], Brian A. Wandell[a], and Peter B. Catrysse[a]

[a]Department of Electrical Engineering, Stanford University, CA 94305
[b]Hewlett-Packard Laboratories, Palo Alto, CA 94304

ABSTRACT

We design and analyze a high-speed document sensing and misprint detection system for real-time monitoring of printed pages. We implemented and characterized a prototype system, comprising a solid-state line sensor and a high-quality imaging lens, that measures in real time the light reflected from a printed page. We use sensor simulation software and signal processing methods to create an expected sensor response given the page that is being printed. The measured response is compared with the predicted response based on a system simulation. A computational misprint detection system measures differences between the expected and measured responses, continuously evaluating the likelihood of a misprint. We describe several algorithms to identify rapidly any significant deviations between the expected and actual sensor response. The parameters of the system are determined by a cost-benefit analysis.

Keywords: Document sensing, misprint detection, printing quality control, digital presses

1. INTRODUCTION

High-speed digital presses are increasingly used to print short runs of books or books on demand. The number of on-demand and short-run printed titles increased almost 8 fold between 2002 and 2008. Today half of the printed titles in the Unites States are printed in short runs.[1] In contrast to traditional presses, short-run digital presses typically print a relatively small number of copies, ranging from one copy to a few hundred copies.

While misprint detection is an important capability in any printing system, it is particularly important in short-run printing. If a print error is caught after 50 misprints in a short-run of 100 copies, the contribution to the final cost is much higher than in a traditional run of 10,000 copies.

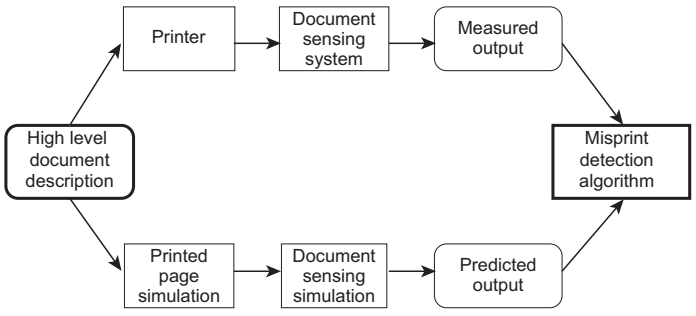


Figure 1. **Overview of a document sensing and misprint detection system.** A high-level document description is sent to the printer, where it is measured by the document sensing system (measured output). The document is simultaneously sent to a printed page simulator and and a document sensing system simulator resulting in a a predicted output. The measured and predicted outputs are compared by a misprint detection algorithm.

Email: pcatryss@stanford.edu, wandell@stanford.edu

Sensors, Cameras, and Systems for Industrial/Scientific Applications XI, edited by Erik Bodegom, Valérie Nguyen, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7536, 75360A · © 2010 SPIE-IS&T · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.840325

Given the historical importance of traditional rotary presses, there exists a large body of prior art regarding misprint detection for such systems.[2,3] Misprint detection is often performed by first establishing a correctly printed copy and comparing this to subsequently printed copies. Given a large print volume, this is an acceptable solution. At the other end of the printing spectrum, consumer-grade printers (e.g., ink jet and laser printers) provide some self-testing capabilities: a specific pattern is printed and the system decides if the printer is working correctly.[4] While a useful way of calibrating and cleaning the printer, this approach does not guarantee that the printed copy is correct. Researchers from Hewlett-Packard present a fully automatic system for misprint detection.[5] This system, however, does not take into account the real-time constraints and computational cost involved in high-speed digital presses. In addition to the aforementioned patent publications, there has also been some academic work on the topic of misprint detection.[6]

In this paper, we propose a high-speed document sensing and real-time misprint detection solution for short-run digital printing presses. The proposed system comprises three parts (see Figure 1). The first part is an (optical) document sensing system that can be embedded into the printing path. The purpose of this system, which operates under fixed illumination, is to acquire information about the rapidly moving printed page. The second part is a forward-prediction system, based on a system simulation, that takes the rasterized page as input and predicts the response of the document sensing system. The third part is a computational misprint detection system which comprises a set of algorithms that compare the predicted and measured signals. The parameters of the system are determined by a cost-benefit analysis.

2. METHODOLOGY

The challenges in implementing the proposed solution include integrating system components capable of measuring printed output at high speed and designing algorithms to rapidly and reliably detect print errors. For the first part, we develop and implement a prototype high-speed, high-sensitivity optical document sensing system that measures the light reflected from the printed page under known illumination. The high-speed requirement means that sensors must use short integration times. In the process, we identify and characterize the performance of suitable system components, including sensors, light sources, optics.

The second system part calculates the expected sensor output given a successfully printed page. The input to the page prediction system is a rasterized CMYK (Cyan Magenta Yellow blacK) dataset representing the page of a document. The CMYK dataset is transformed into a reflectance description of the printed page. We use the measured reflectance spectra from a representative set of uniform color patches (CMYK calibration set) printed by a digital press to implement this calculation. Provided a known illumination source, we compute the scattered light captured by the document sensing system using ISET (Image System Evaluation Toolbox[7–9]). ISET relies on radiometric scene information (from the printed page) and device specific characteristics (from the lens and sensor) to predict the system response. The real-time constraints of the system require us to derive pre-computed lookup tables that speed up several steps in the ISET calculation. The document sensing system simulation enables us to calculate the expected system response for any printed page and to understand the consequences of changing system characteristics.

The third part of the proposed misprint detection solution consists of algorithms that compare the expected and measured sensor responses. These algorithms have to be efficient given the high-speed throughput of the system. Rather than comparing page content point-by-point at high temporal and spatial resolution, we extract general features and compare those between the measured and predicted signals (e.g. mean signal level and large deviations in signal levels). We discuss two classes of algorithms for detecting potential misprints: a theoretical approach based directly on error statistics (Section 4.3), and a learning approach (Section 4.4). Both algorithms are based on a study of the deviations between the predicted and sensed images.

3. DOCUMENT SENSING SYSTEM

In this section, we describe a prototype document sensing system and its characterization, a method for converting a high-level document description into a physical description of the light scattered from the printed page, and an ISET-based procedure to create an efficient computational simulation that generates the predicted output.

3.1 Optical detection system

The key part of the document sensing system is a high-speed optical detection system. Our prototype system consists of an illumination source, a high-quality imaging lens (f=60mm, F/2.8), and a charge-coupled device (CCD) line sensor (2048 pixels per line). The lens has diffraction-limited performance on-axis. The CCD sensor has a linear array of pixels and features time-delay integration (TDI) with 96 lines. The multiple lines provide built-in pixel binning to improve signal quality when the sensor runs at a high-speed (2770 lines per second).

To evaluate the prototype system, a printed page is placed on a rotating drum. The speed of the rotating drum is adjusted to emulate different page printing speeds. For each print speed, the readout rate of the CCD line sensor is synchronized to the drum. The drum achieves speeds from 1 to 5 pages per second, which corresponds to typical print rates of 60 (full-color) to 300 (monochrome) pages per minute for high-speed digital presses.

The prototype system is also used to measure the parameters for the model of the document sensing system, which is needed for page prediction. These parameters include geometrical parameters (pixel size, fill-factor), electrical properties (conversion gain, noise) and optoelectronic parameters (spectral quantum efficiency) (Figure 3). They are summarized in the table in Figure 3. Finally, we model the known light source by its spectral power distribution ($D65$) and luminance (cd/m^2).

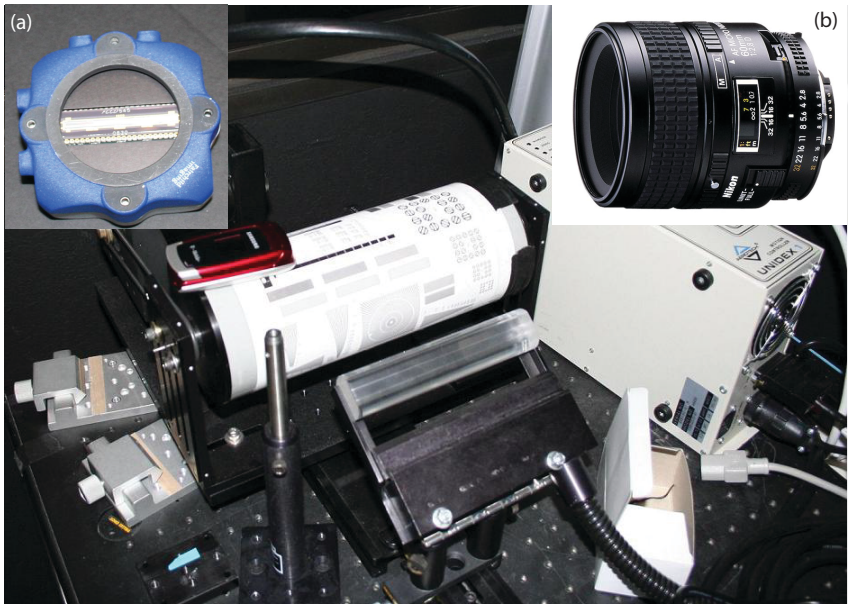
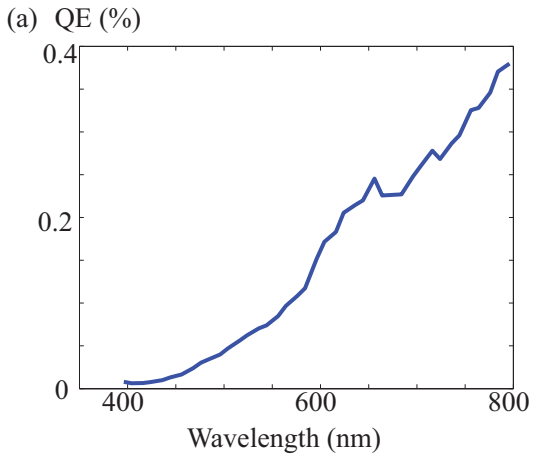


Figure 2. **Prototype system** consisting of a high-speed CCD line sensor, a high-quality imaging lens and a rotating drum representing a moving web. Inset show (a) the CCD sensor and (b) the imaging lens

3.2 Page prediction system

We create a physically accurate description of the prototype system by modeling its individual components in ISET. The component model for the optics simulates the optical path from the object plane (printed page) to the image plane where the sensor is located. For the optics model, we assume a diffraction-limited model based on the f/# of the imaging lens. The model for the sensor component simulates the electrical path inside the sensor. The sensor model is based on the characterization data that we acquired in the lab supplemented by manufacturer specifications of the CCD line sensor (see Figure 2). The ISET simulation of the optical detedtion system is an essential component of the page prediction system; it also enables us to experiment with various system design options (Section 5.3).

In addition to the document sensing model, we have to convert the page description, typically given as a rasterized CMYK dataset (an array of CMYK values), into a radiometric description of the light scattered from the page. A key step in this process is deriving the printed pages surface reflectance from the page description (see Figure 4). We make



(b)

Read noise	1.1 DN (8-bit)
Dark current	23.9 DN/ms (dc)
Dark signal non-uniformity	5.9%
Full well capacity	200,000 e-
Dynamic range	3,000
Pixel conversion gain	3 V/e-
Quantum efficiency (650nm)	0.22
Pixel size	13 x 13 (um)
Fill factor	100 %
Photo response non-uniformity	5.9 % (TDI = 96)

Figure 3. **Measured system parameters.** (a) Spectral QE of CCD lines sensor. (b) Electrical and geometrical parameters.

this estimate by measuring the reflectance functions of a sample array of 1338 small color patches. These samples were measured at 36 wavelengths (380 nm to 730 nm at 10 nm intervals). We predict the reflectances of CMYK values that are not in the CMYK calibration dataset by interpolating the measured sample data. This conversion step has to be carried out quickly. With this in mind, we evaluate and compare two different interpolation methods.

3.3 Page description (rasterized CMYK dataset) to surface reflectance

One approach is to interpolate the reflectance data, for each CMYK value in the dataset. For demonstration purposes, we use linear interpolation, but other (more complex) interpolation methods are applicable as desired. For every rasterized entry in the CMYK dataset of the page description, we find its N closest neighbors in the CMYK calibration set and compute the interpolated surface reflectance value for every wavelength considered.

A second approach is to fit a simple model that predicts the spectral surface reflectance for the CMYK dataset values. A log-linear model provides a good approximation to the measured data when we fit the model separately for each wavelength. The formula for the log-linear model for each wavelength is based on finding the vector $a(\lambda)$ that minimizes

$$min_a(\sum_i (r_i(\lambda) - log(a^T v_i)^2) \tag{1}$$

where r_i^λ is the reflectance of the i^{th} sample at λ, and $v_i = (c_i, m_i, y_i, k_i, 1)$ is the i^{th} CMYK representation. Once the minimization is solved, the estimated reflectance for any CMYK sample, v, is computed as $log((a(\lambda)^T v)^2)$. This approach results in a small fitting error for our CMYK calibration set. It is also faster than the linear interpolation approach and requires very little memory. The simulations below are performed using the log-linear model.

3.4 Reflected light to predicted sensor sensor responses: Computational efficiency

Given the system illuminant ($D65$) and the spectral reflectance at each point of the printed page, we can calculate the spectral radiance ($W/sr/nm/m^2$) of the scattered light. Given the optical configuration of the system, we can compute the spectral irradiance ($W/nm/m^2$) of the light incident on the sensor pixels. With the optics and sensor properties characterized from the prototype system, we use the ISET toolbox to compute the predicted sensor signal.

There remains one important difficulty: the complete system simulation is physically accurate but requires a significant amount of computation. This computational cost conflicts with the real-time requirement of misprint detection. For a real-time system, it is essential to speed up the computation; we can achieve a much faster computation by understanding that the system computation can be divided into two parts. One part provides the mean sensor response to a CMYK sample; and the second part provides the system noise. By repeating the phsycially accurate ISET-based system simulation a great

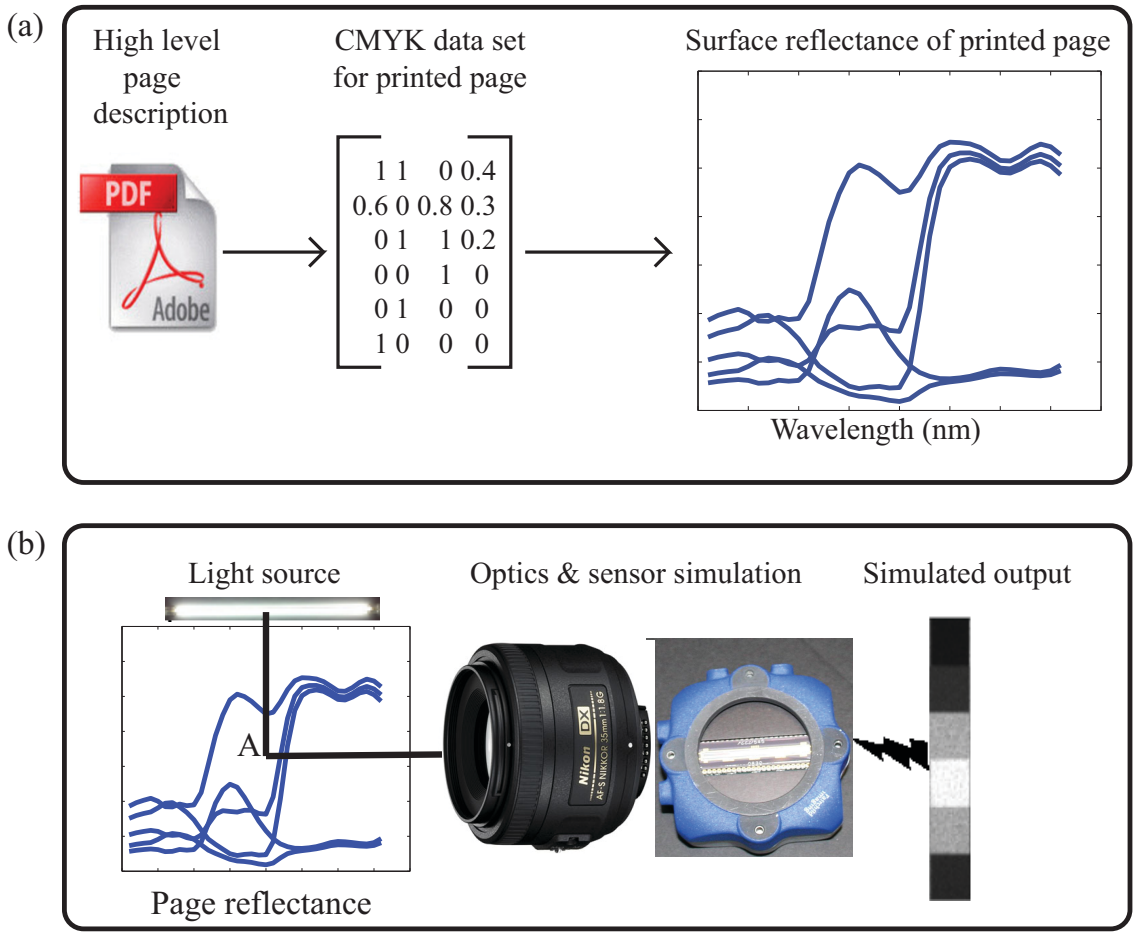


Figure 4. **Document sensing system - real-time simulation.** (a) Conversion from high-level document description to printed page reflectance. (b) ISET-based simulation from surface reflectance to sensor output. A:Multiplication in the wavelength domain of the light properties with the reflectance of the page surface.

number of times, we can compute two look-up tables: the first table (the mean table) stores the average value of the sensor output for each CMYK sample; the second table (the noise table) stores the statistical properties of the deviations from this average. This deviation from the average signal reflects the contribution of all the noise sources, including shot noise inherent to the optical signals and sensor noise added by the detection system. Computing the mean and noise tables is equivalent to calibrating the sensor system. The construction of the look-up tables is a significant computation, but it only has to be done once.

In an ideal solution, we would store the mean for every CMYK sample and use a set of measured samples to determine the complete noise distribution. A further simplification is possible for this particular sensing system; our experiments show that apart from issues of sensor saturation the measured responses are well-fit by an additive, zero-mean Gaussian distribution. Hence, it is efficient to store only the standard deviation in the noise table.

We can now simulate the sensor response to a printed page efficiently. The complete efficient method consists of using the mean table to derive the mean sensor response to a CMYK value, and using the noise table to store the standard deviation of the noise to know the noise distribution. The predicted output is computed by only using the mean table as the mean signal is the most likely signal (our best guess) - We willl use the noise table later-. The significance of this method is that it provides a way to simulate the sensor output to any printed page, including pages with specific types of printer errors, without having real examples of those pages at hand.

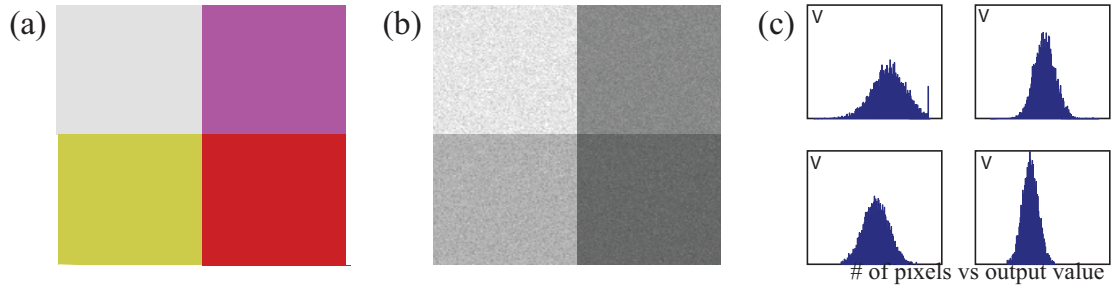


Figure 5. **Efficient page prediction using pre-computed look-up tables.** (a) Example color patches, represented in the CMYK color space. (b) Sample sensor output values. (c) Histograms of the sensor output. Distributions for different colors are close to Gaussian, but each has their own mean and standard deviation.

4. MISPRINT DETECTION

We declare a misprint when the measured response to a printed page differs significantly from the expected response. In the page prediction system, the expected response to a page is calculated using the mean table. The likelihood of observing a sensor response given no misprint is equivalent to assessing the likelihood of the observed differences. Hence, the first step in misprint analysis is to express the difference between the observed and expected sensor response. It is convenient to do this using a method that normalizes the size of the errors, and we describe how to create this representation in the next section.

The likelihood of observing the deviation can be computed in two ways. First, if there is a closed-form expression for the noise distribution, we can calculate the likelihood directly. This is possible for the prototype system analyzed here, and we describe the direct calculation in Section 4.3. Second, without a closed-form expression for the noise it is still possible to use a learning-based procedure to develop a misprint classifier. We describe this approach in Section 4.4.

4.1 Normalized error map

For the prototype system, and many other likely systems, the error distribution will be close to a zero-mean Gaussian distribution with a standard deviation that varies with the CMYK value. The standard deviation depends on the printed CMYK value because of several factors, including the Poisson noise in sensor quantum catch (shot noise) and signal-dependent noise in the sensor (e.g., photo-response non-uniformities).

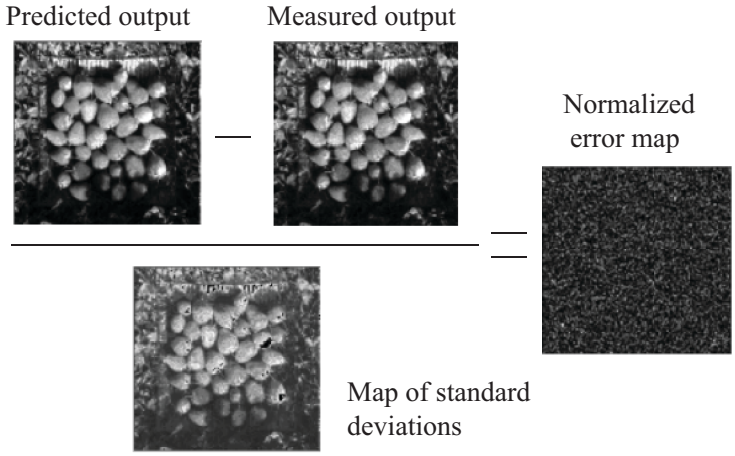


Figure 6. **Normalized error map.** The difference between the predicted and measured sensor signals is normalized by the standard deviation of the noise on a pixel-by-pixel basis. This normalized error map serves as an input for the misprint detection algorithm.

4.2 Pre-processing of the sensing system output.

When the noise distributions at different CMYK levels are identical, apart from the standard deviation, it is convenient to create a normalized error map (see Figure 4.1). This normalized map represents the difference between the expected and observed signals divided by the noise standard deviation (z-score). Specifically if the measured signal is $(x_{i,j})$, and the expected signal is $(y_{i,j})$, the normalized map is

$$z_{i,j} = \frac{x_{i,j} - y_{i,j}}{\sigma_{i,j}} \tag{2}$$

We use the normalized error map, $z_{i,j}$, as the starting point for both misprint detection methods described below.

4.3 Misprint detection: Statistical analysis of the noise

We can calculate the likelihood directly if we have a closed-form expression for the noise distribution. Under the null hypothesis *the print is correct* the normalized error map values, z, should be normally distributed with zero mean and unit variance. We can test the null hypothesis using a Pearson χ^2-test[10,11] using the test statistic, S,

$$S = \sum_n \frac{(O_n - f_n)^2}{f_n} \tag{3}$$

In this formula the z-scores are partitioned into n different intervals. The values O_n are the number of observed z-scores in the n^{th} interval and f_n is the expected number of such values under the null hypothesis. The test statistic asymptotically approaches a χ^2 distribution as the number of samples becomes large.

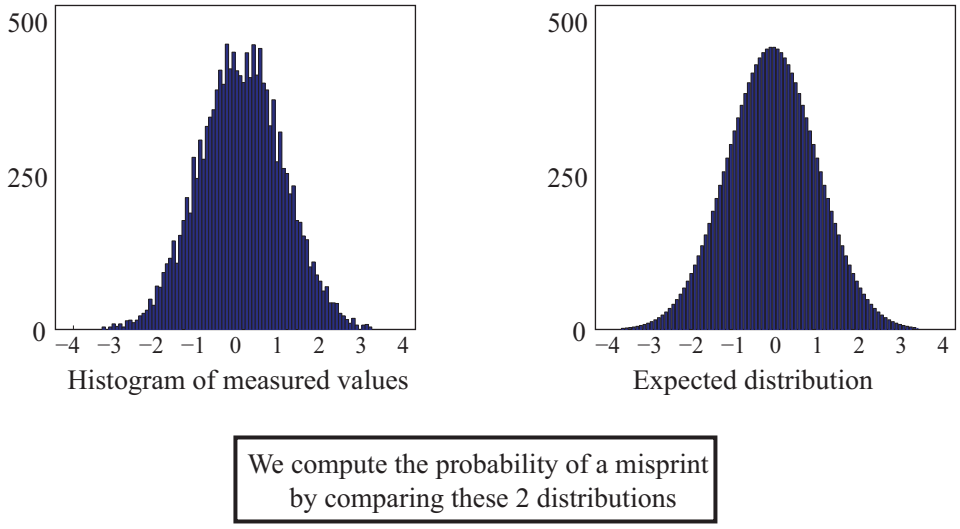


Figure 7. **Theoretical calculation of the likelihood of a misprint.** Comparison of the histogram of the measured differences, and thehHistogram for the modeled distribution (Gaussian here). We then compute the probability of a misprint by comparing these 2 distributions.

For the statistic S, we can calculate a p-value $p = 1 - g(S)$, using the cumulative distribution function g of the χ^2 distribution with $n-1$ degrees of freedom. The p-value gives us the probability of obtaining a result that is at least as extreme as S. We then set a threshold on this p-value to decide if we have a misprint or not.

Our tests show that misprinted page images have an extremely low p-value. We can therefore set a very low threshold for p before rejecting the hypothesis of a correct print. This threshold can be chosen to minimize a cost function accounting for the financial cost of a misprint and a false alarm.

The direct method is very versatile and can be adapted to other error models, as long as we can normalize the error map. One limitation of this method is that it relies on the assumption that the noise distribution is known to a high degree

of accuracy. When there are deviations between the noise in the actual system and the noise in the system model, e.g., as a result of a mis-calibration of the noise, performance can suffer greatly.

4.4 Misprint detection: Learning method

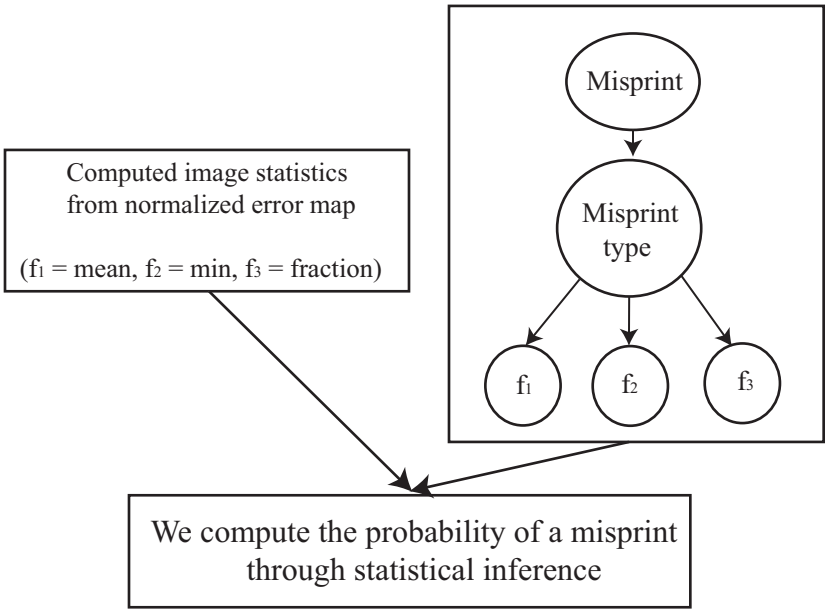


Figure 8. **Statistical learning method for misprint detection.** Statistics are computed from the normalized map of differences. We then compute the probability of a misprint through statistical inference.

To improve the robustness of misprint detection to calibration and modeling errors, we developed a learning-based algorithm (Figure 8). The learning-based algorithm relies on the same simulator used in the theoretical approach. It differs in that it is less dependent on the accuracy of the model. Hence, it is much more robust to errors in the mean and noise tables.

We use a statistical model[12] that tries to capture the general behavior of the system. The statistical model consists of a number of random variables that describe the system. The variables are connected through a Bayesian network (see Figure 8, right panel) that describes the conditional relations between these random variables. If we know the conditional distributions of probability of each variable, given the others, we can infer the posterior probabilities from the observations. We consider 5 random variables: a binary Misprint random variable ($M \in 0;1$ where 0 represents a correct print and 1 a misprint), a Misprint type random variable ($M_t \in global, local, correct$), and $f = 3$ image statistics.

If the Misprint type is known, it is trivial to know if a misprint occured as $M = 0 \Leftrightarrow M_t = 0$. To know M_t we need the conditional distributions of probability of the three image statistics f. We do not know these a priori so we have to learn them with the following algorithm:

1. We can use our simulator on an image database, and introduce misprints on images.
2. Next, we compute these features for the correct image and for the misprinted image.
3. We can then quantize these features in a finite number of values and count for each value how many times it happened.
4. Finally, we divide by the total. This gives us $P(f_i|M_t)$. It is safer to apply Laplacian smoothing, i.e., start with the counts at 1 instead of 0 to avoid zeros probabilities in the final table, which would give impossible cases.

To compute these different image statistics, we assume that the noise distributions for a correct print are completely described by their standard deviation (e.g. a zero-mean normal distribution). The first step of the algorithm takes the preprocessed data and produces a normalized error map (see section 4.1).

Next, we select non-overlapping k-by-k windows, and average the errors in each window. For f image statistics and n^2 pixels, this allows us to reduce the computation from fn^2 to $(1+f/k^2)n^2$ operations. Alternately, we can also use overlapping windows for a $(f+1)n^2$ cost. Finally, we summarize the behavior of the error map by computing the image statistics. In our implementation, these statistics are the average value, the minimum value, and the fraction of errors below 0.675 standard deviations (which represents a 0.5 probability for a normal distribution, with $\sigma^2 = 1$).

Different misprints can have a very distinct effect on the chosen statistics. For example, misprints that have an effect on the whole page (e.g., a missing color plane) have a big effect on the mean statistic, while those that affect a small area of the page do not change the mean statistic much. It is therefore important to distinguish between global misprints with an effect on the whole page, and the local misprints, with only a local effect. We found that misprints within each of these two categories tend to have a similar effect on the image statistics.

We assume for the statistical model that the $P(f_i|M_t)$'s are independent. This is not entirely true. As the misprints within one of the categories have a similar effect on the statistics, this assumption becomes more likely. This gives us a quite general method to detect misprints. Because it is a learning based algorithm, the final parameters depend on the training set that has to be representative of the possible inputs. It is also biased towards the misprints that we implemented. Nevertheless this bias is probably not too strong as we are summing up the image in three numbers only, and we have a large variety of misprints. It is therefore less likely to encounter a misprint that does not behave like any implemented misprint regarding our three features.

The learning-based method is more robust to look-up table errors than the statistical model. For example, the statistical approach fails if the voltage swing estimate is inaccurate, but the learning based method correctly adapts. The improved robustness comes from the fact that this model learns the error distribution and thus errors in our assumptions are immaterial. Because the algorithm compares sets of deviation between correct images and misprinted images, we do not need to explicit this model, we are just assuming that, for any two random deviations X and Y, iand for any (x,y):

$$P_X(X > x) = P_Y(Y > y) \rightarrow \frac{x}{\sigma(X)} = \frac{y}{\sigma(Y)} \tag{4}$$

In particular this is clearly true for 0-mean normal distributions: $P_X \sim N(0, \sigma(X)^2)$ as in our simulation. We can also see that a proportional error on σ (e.g. $\sigma_{\text{measured}} = \alpha\sigma$) has no influence. More generally, if we assume a small error on the σ's: $\sigma_m(X) = \sigma(X) + \varepsilon_X$ then:

$$\left|\frac{x}{\sigma(X)+\varepsilon_X} - \frac{y}{\sigma(Y)+\varepsilon_Y}\right| \sim \frac{|x|}{\sigma(X)}\left|\frac{\varepsilon_X}{\sigma(X)} - \frac{\varepsilon_Y}{\sigma(Y)}\right| \tag{5}$$

Therefore, with $\varepsilon_X \ll \sigma(X)$ we stay very close to our assumption (see equation 4).

5. RESULTS

We now demonstrate the use of the misprint detection system. We prepare a test set of pages to be printed. For each page in the test set, we randomly select whether there is a misprint or not. If a misprint is selected, we also choose a type of misprint. There are misprint twelve types ranging from a missing color plane to small local blur. The misprint detection system is evaluated by performing a series of simulations and by applying the system to every image of a printed page in the test set. The algorithm computes the probability of a misprint; we set the uncaught misprint rate and the false alarm rate by varying the misprint decision threshold and comparing the result to the ground truth.

A learning set is required for the learning-based method. The learning set (1000 pages), like the test set (300 pages), is composed of full page images. Both sets include pages with a wide array of colors and misprint types. For a commercial implementation, the training and testing set should be adapted to the kind of printing done, for instance pages with mostly text.

5.1 Cost-benefit analysis

The goal of the system is to minimize the cost of possible misprints. Here, we show to derive a cost function (Equation 7) and set the threshold to minimize this cost function.

The cost per page due to misprints can be estimated as:

$$C = (1-r)\,c_f F + r\,c_m M \tag{6}$$

Minimizing this cost function C is equivalent to minimizing c defined by:

$$c = F + \alpha M \tag{7}$$

where r is the misprint rate, $F = P(\text{Predicted misprint}|\text{correct print})$ and $M = P(\text{Predicted correct print}|\text{misprint})$. c_m is the cost of an uncaught misprint, c_f is the cost of a false alarm, and $\alpha = r\,c_m/(1-r)\,c_f$. Minimizing this function is done by varying the misprint threshold, then computing the values of F and M for each value of the threshold. Picking the threshold with the lowest associated cost will then minimize the cost function.

We report the cost-benefit results in tabulated form. The values in these tables, (e.g., Table (1), are proportional to the total cost. We can use these values to compare the ratios of the costs between different methods and systems. The tables compare the cost of different methods as a function of the value of α, which represents the relative costs of failing to catch a misprint and generating a false alarm. Values should be compared for a common level of α. The tables indicate the lowest cost in each column by bold font. Note that values between columns represent different criteria levels for a system and the cost always increases with α (see Equation 7).

Table 1. **Cost-benefit analysis.** Values of c for 1 sensor

α	0.001	0.01	0.1	1	10	100
1 sensor	$3.4 \cdot 10^{-4}$	0.0034	0.0338	0.1366	0.3821	1.0000

5.2 Example analysis

To illustrate how to use these tabulated results, we compute an example using the values in Table 1. Consider a system where: $r = 0.01$ (There is a misprint every hundred page), $c_f = 0.01$ cent and $c_m = 0.1$ cent. For this system $\alpha \sim 0.001$. At this level of α, $c = 3.4 \cdot 10^{-4}$, and the expected cost per page is $C = (1-r)c_f c = 3.4 \cdot 10^{-4}$ cent. If we print 10000 pages, the expected cost is $C_{10000} = 3.4$ cents. We can compare this cost with the cost when the system is turned off. When the system is off, the cost is the rate of misprints times the cost of each misprint. Using the same values, 10000 pages and a misprint rate of 0.01, the expected cost of misprints is $\hat{C}_{10000} = 10$ cents. In this example, the system reduces the cost of misprints by a factor of three.

5.3 System design using cost-benefit analysis

We can use the ISET system simulation to evaluate the cost-benefit of system design changes. For example, we can evaluate the value of adding an additional sensor, improving the noise properties, or calculating for a different number of misprint types. The simulation results conform to our expectations: two sensors give a better result that one sensor, a noisy sensor is less efficient, and a model with 3 misprint types outperforms a model with only 2. The cost-benefit results for these configurations are shown in Tables 2, 3 and 4.

Table 2. **System evaluation.** Comparison of cost c for 1 and 2 sensors

α	0.001	0.01	0.1	1	10	100
1 sensor	$3.4 \cdot 10^{-4}$	0.0034	0.0338	0.1366	0.3821	1.0000
2 sensors	$\mathbf{2 \cdot 10^{-5}}$	**0.0002**	**0.0020**	**0.0204**	**0.1562**	**0.7809**

Adding another sensor leads to a significant increase in value (Table 2). The second sensor is simulated with same noise parameters as the first, but a spectral QE that complements that of the first (other parameters were the same). Depending

on α, the misprint cost is reduced by as much as a factor of 17, and for any α level using a second sensor reduces costs. Even when $c_m/c_f = 100$, the cost is reduced by 30%. This reduction in misprint cost should be compared with the expense of adding a sensor. These costs take two forms: One is the cost of the sensor, which is not very high; the second is the additional computational cost which is significant. It is a very good approximation to say that the computational cost is linear in the number of sensors, so adding a second sensor doubles the computational power needed to run the algorithm. Moreover we also need to calibrate different look-up tables for each sensor.

Table 3. **System evaluation.** Comparison of cost c for a 1 misprint type and a 2 misprint types model

α	0.001	0.01	0.1	1	10	100
1 misprint type model	$6.7 \cdot 10^{-5}$	$6.7 \cdot 10^{-4}$	0.0067	0.0404	0.3025	1.0000
2 misprint types model	$\mathbf{2 \cdot 10^{-5}}$	**0.0002**	**0.0020**	**0.0204**	**0.1562**	**0.7809**

Table 3 analyzes the effect of separating local and global misprints in the statistical calculations (see subsection 4.4, and Figure 8). The model with two misprint types slightly outperforms the simpler model: for an α level of 1, the cost of misprints is reduced by a factor of 2. This improvement comes almost for free. The additional computations are negligible compared to simulating the sensor, requiring a little more memory to store the learned parameters.

Table 4. **System evaluation.** Comparison of cost c for several noise levels

α	0.001	0.01	0.1	1	10	100
100% noise	$\mathbf{7.2 \cdot 10^{-5}}$	$\mathbf{7.2 \cdot 10^{-4}}$	**0.0072**	**0.0545**	**0.1802**	**0.7760**
200% noise	$1.1 \cdot 10^{-4}$	0.0011	0.0112	0.0939	0.4711	0.8403
300% noise	$1.4 \cdot 10^{-4}$	0.0014	0.0137	0.0869	0.6157	1.0000
400% noise	$3.3 \cdot 10^{-4}$	0.0033	0.0326	0.2811	0.9730	1.0000

Table 4 shows the effect of sensor noise for a two-sensor system. A 100% noise corresponds to the noise parameters of the sensor described in Table 3. Naturally, a noisier sensor is less accurate and increases the cost due to misprints at each α level. The key problem is that as the noise level increases, distributions for different CMYK values overlap more. Consequently, a pixel value becomes likely for several different colors, making much more difficult to distinguish between a correct print and a misprint. Here the cost-benefit analysis provides guidance on how much we should value noise reduction in the sensor.

6. CONCLUSION

The system described here efficiently detects misprints in a high-speed digital press. The procedure uses software to model the system and create look-up tables needed to implement an efficient computational process that predicts the sensor output. Depending on the accuracy of the simulation, and the noise distribution, we can detect misprints using either a theoretical approach or a learning-based method that is robust to simulation and calibration inaccuracies. The system allows the user to calculate trade-offs between cost and benefit for changes in system hardware and statistical algorithms.

ACKNOWLEDGMENTS

This work was supported by a 2008 HP Labs Innovation Research Award from the Open Innovation Office at Hewlett-Packard Laboratories.

REFERENCES

[1] "Bowker industry report, new book titles & editions, 2002-2008," (2009). http://www.bowker.com/bookwire/IndustryStats2009.pdf.

[2] Stringa, L., "Method for automatically checking the printing quality of a multicolor image," (2001). US Patent 6,301,374.

[3] Honma, M., "Image forming apparatus and misprint detection method," (2008). Patent Application US20080260395.

[4] Sheinman, Y. and Weksler, M., "Continuous ink jet printing apparatus and method including self-testing for printing error," (1999). US Patent 6,003,980.
[5] Kisilev, P. and Ruckenstein, G., "Print defect detection," (2009). US Patent 7,519,222.
[6] Westra, R., Shippen, J., and Freear, N., "Printing quality control using template independent neurofuzzy defect classification," 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT) (1999).
[7] Farrell, J. E., Xiao, F., Catrysse, P. B., and Wandell, B. A., "A simulation tool for evaluating digital camera image quality," in [*Proc. SPIE*], **5294**, 124 (2003).
[8] Farrell, J. E., Ng, G., Ding, X., Larson, K., and Wandell, B. A., "A display simulation toolbox for image quality evaluation," *Journal of Display Technology* **4**, 262–270 (2008).
[9] Maeda, P. Y., Catrysse, P. B., and Wandell, B. A., "Integrating lens design with digital camera simulation," in [*Proc. SPIE*], **5678**, 48 (2005).
[10] DeGroot, M. H. and Schervish., M. J., [*Probability and Statistics*.], ch. 9, Addison-Wesley, 3rd ed. (2002).
[11] Plackett, R. L., "Karl pearson and the chi-squared test," *Int. Stat. Rev.* **51**, 59–72 (1983).
[12] Bishop, C. M., [*Pattern Recognition and Machine Learning*], ch. 8, Springer (2006).