Closed Loop Color Control in the Work

John Seymour*, Steve Tiltman*, and Adam Nelson*

Keywords: Color, colorimetry, inking

Abstract

Fifteen years ago, closed loop color control systems were introduced for commercial web offset presses. These systems soon became standard equipment. This generation of color control systems utilized color bars. The ink keys were adjusted so as to maintain proper color characteristics of the patches in the color bar.

The advancement of technology has just recently made the next generation of color control systems possible. These systems do not rely on color bars but rather control color based on measurements of the printed images.

Controlling color bars is relatively easy to understand, at least in principle. A measurement is made of a patch with a single color of ink. If the color is too weak, inking levels are increased. If the color is too strong, inking levels are decreased. Gray balance control is a bit more complicated, but the strategy is similar. There are two features of these control systems that simplify the task. First, the color bars are a known shape and size. Second, the color bars are a known that do not change from one job to the next.

For Color Control In the Work (CCIW), the control system must be able to adapt to whatever is printed. Such a system must be able to decide how to adjust color for any ink combination. How this might be done is not entirely obvious. For this paper, I have reviewed a number of patents to understand a number of ways that CCIW might be performed. These methods are described. I also show the results of tests performed on a CCIW system that we have developed.

Why Is It Necessary to Control the Color of the Work?

Traditional online color control systems make use of color bar measurements to adjust the inking levels on the press. The underlying assumption is that having the correct colors in the color bar ensures having the correct color in the work.

The success of color bar control systems shows that this is a reasonable assumption. However, at least one early paper had predicted that this would not work. Mason (1987) stated that “solid color bars are an unreliable source of information about image color variation after the run is in progress.” Three

*QuadTech

recent TAGA papers (Breede, 2007; Sigg, 2007; and Siljander, 2001) reiterate these earlier concerns. All of the papers have shown that circumferential variation within an ink key zone may be up to 0.10D. Although the color measurements of the color bar itself might be well within tolerance, a hypothetical color bar printed elsewhere on the form might be well outside of tolerance.

Basically, the color bar only approximates the color of the work. Perhaps this is why press operators often spend time fine-tuning color after the color control system has gotten the color bars within tolerance.

CCIW systems are just becoming available. The desire to eliminate this fine-tuning step has not been the only driving force for this type of system. Instead, a need to eliminate color bars has driven this technology.

Most commercial work (catalogs and magazines) is trimmed when the book is assembled. This means there is an area between impressions that the final customer never sees. This has traditionally been a hiding place for color bars. There are other markets, on the other hand, that don't have this luxury. Newspapers and free-standing inserts (commercial work that is not bound, for example, inserts in a newspaper) do not have trim. Placing a color bar on the front page of a newspaper is undesirable because they are distracting. Hence, the newspaper market—and not higher-quality publications—is driving the development of CCIW systems.

CCIW—Color Control in the Work

A Few Methods

For this paper, I have surveyed the patent literature to find suggestions about how one might construct a CCIW system. I summarize four basic approaches: 1) Finding color bars in the work, 2) CMYK measurement through RGB color separation, 3) CMYK measurement through RGB+I color separation, and 4) CIELAB optimization.

Finding Color Bars in the Work

The first approach I will call "finding color bars in the work." An existing (or slightly modified) point measurement device could make measurements of selected spots in the image and base its control moves on these measurements.

There are a number of color control patents that have some verbiage to this effect. The patents generally cover some form of color control system that was designed for control from a color bar. I suspect that the patent writers wanted to make sure that they weren't limited to control from color bars, so they added language to broaden the patent.

The problem is that the method assumes that it is possible to find suitable color patch substitutes in the actual work. Figure 1 shows (on the top) an example of an image where a color control system might do solid ink control from the image. But this image was actually hard to find! The image at the bottom is more typical. In the image on the bottom, there are no areas where one can find solid cyan, magenta, yellow, or black inks all by themselves.

Figure 1.

There is an underlying paradigm here. Since we use point measurement devices to measure the color of a color patch, we should use point measurement devices to measure the color of an image. I argue that this is an outdated paradigm.

CMYK Measurement through RGB Color Separation

The second approach I will call "CMYK measurement." In this section, the measurement is done through RGB color separation. Essentially, this amounts to measuring how much of each ink has been put down in a region and then comparing this to a target amount of ink.

This approach uses a standard RGB camera. A picture is taken of the web, so that one has RGB values for each location on the web. These RGB values for each pixel are converted to CMYK values. Each of the C, M, Y, and K values is a measurement of how much of each of the inks are present at that pixel.

Next, all the C values are added together to get a measure of how much cyan ink there is in that entire region. The sum of the cyan values is compared against a target sum for the cyan values in order to decide on a cyan ink move. The other three inks are handled in the same manner.

There is a bit of a misunderstanding embedded in this approach. There is a well-known tool called "color separation" which converts from RGB to CMYK, so it is natural to think that we can use color separation to convert RGB values from the camera into measurements of CMYK. There is, however, a subtle distinction between color separation and what is needed for CCIW.

In traditional color separation, a program like Photoshop determines *one particular* combination of CMYK values that could create a particular set of RGB values. In principle, any combination which renders the appropriate RGB values is acceptable. Photoshop may choose to mix CMY to make a dark color, or may instead rely on black ink.

CMYK measurement as needed for CCIW, on the other hand, has a more constrained requirement. CMYK measurement does not seek *some* CMYK combination that *may have* produced that color, but rather seeks *the particular* CMYK combination on the printed piece that produced the RGB value measured from the web by the camera.

Since there are four inks, and only three RGB values, the transform from RGB to CMYK is an under-determined system of equations. This is analogous to attempting to solve three equations for four unknowns. In general, there are numerous combinations of CMYK that could produce a given RGB combination.

To illustrate the distinction between color separation and CMYK measurement, I used a commercially available RGB scanner to scan the following image from a newspaper. Shown in Figure 2, the image I scanned has a solid cyan triangle with overprints of magenta and black halftones. There is no yellow ink. Since the printing is quite clearly out of register, the location of each of the three inks is apparent.

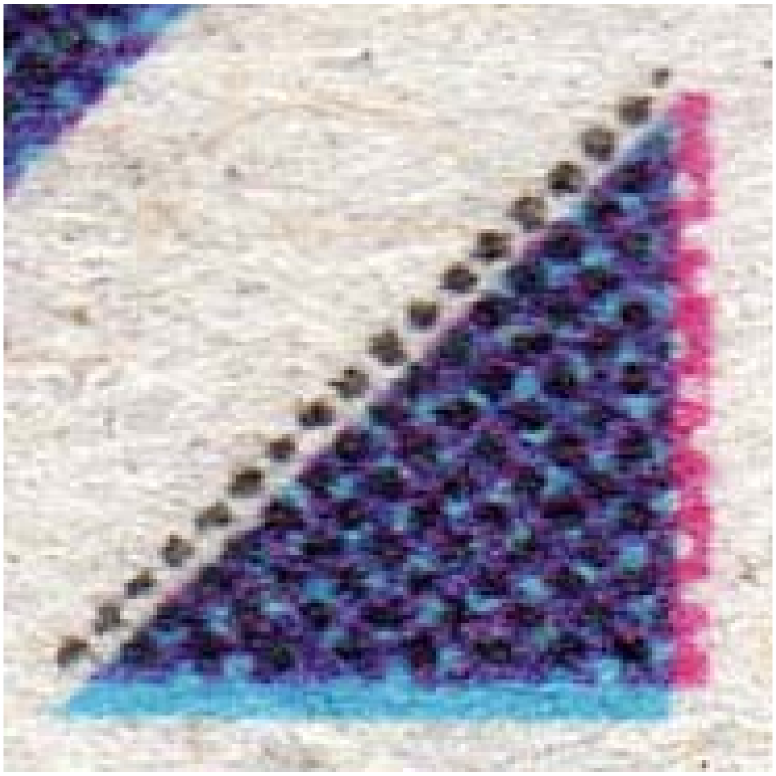
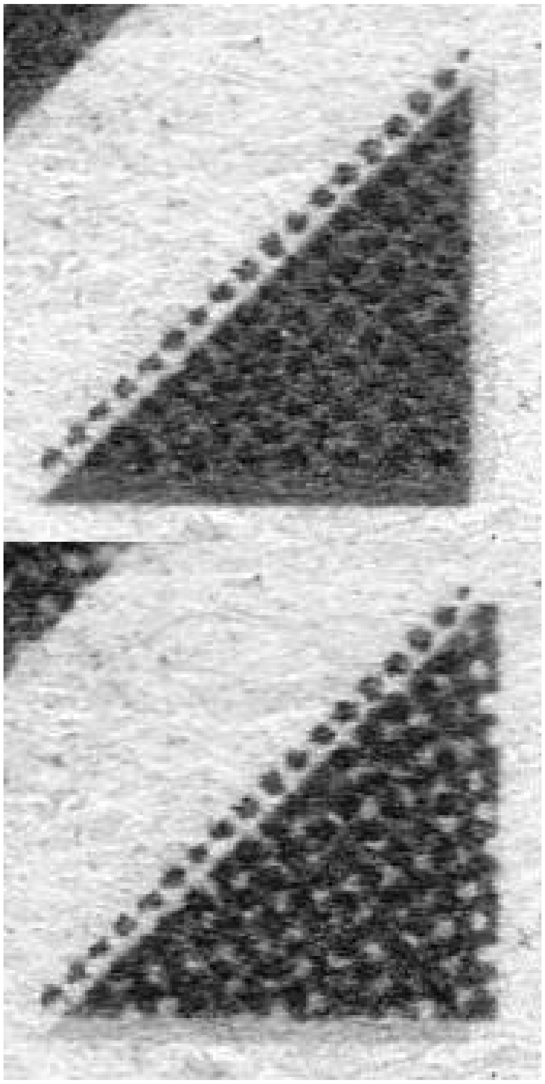
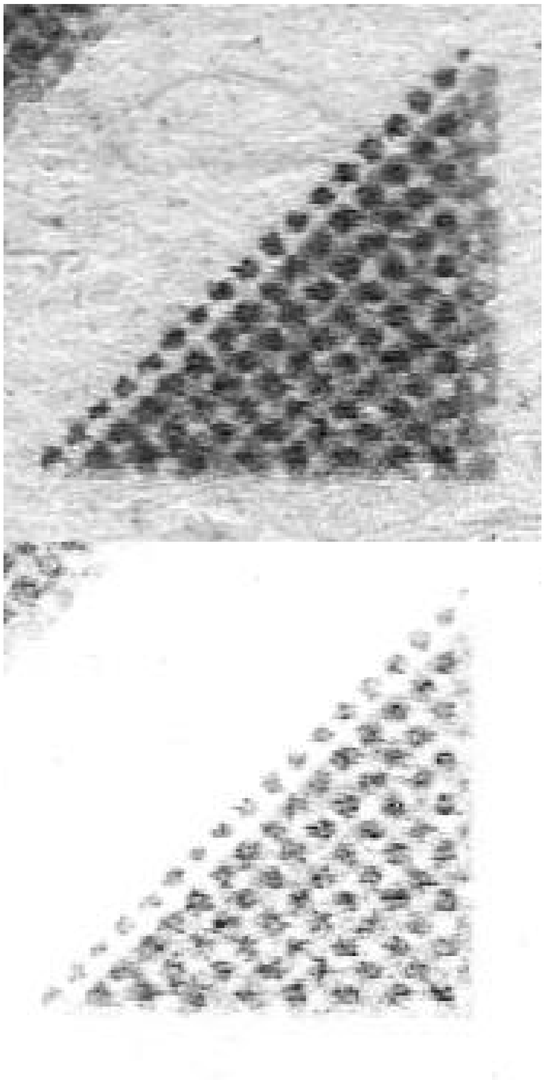


Figure 2.

Color separation was performed on this RGB image using Photoshop with the default settings. This resulted in the four images shown in Figure 3.



Cyan (top), Magenta (bottom)



Yellow (top), Black (bottom)

Figure 3

Several problems of the separation bear some comment. The magenta separation shows some residual cyan ink. Cyan ink typically has some unwanted absorbance that makes it look as if it is magenta ink. As a result, the CCIW algorithm will believe that there is more magenta ink than there actually is and will incorrectly reduce the magenta ink.

Similarly, the yellow image shows some residual magenta ink and black ink. Thus, the CCIW algorithm would believe that there is yellow ink where there should be none, and will incorrectly reduce the yellow ink as well.

The most salient observation, however, is that halftone dots of black ink have shown up in all four images. Once again, the CCIW algorithm will incorrectly conclude that there is too much cyan, magenta, and yellow ink.

For proper CMYK measurement, the black dots should show up strictly in the black image. Oddly enough, the actual black dots are least distinct in the black image. As a result, CCIW will interpret this as insufficient black ink. Black ink will be incorrectly increased.

Thus, regardless of whether there is too much or too little of any of the inks, in this example CCIW will think that there is too much CMY and not enough K.

At first one would be tempted to say that the software performed very poorly. To make the system work, we just need to improve the color separation software. The software, however, performed quite well at the job that it was originally intended to do. The real problem is one of ambiguity and cannot be solved.

It is clear that the standard color separation technique is not suitable for measuring amounts of CMYK for CCIW. Techniques similar to color separation have been proposed (Mason, 1885; Stanford, 1994; Van Holten et al., 2006). Mason uses color mixing models. Stanford uses inverse Neugebauer equations. Van Holten et al. use an incoherent combination of lookup tables, neural networks, a special color space known as “Lag,” and ratios of unspecified quantities. But they all suffer from the same shortcoming: trying to derive four pieces of information from three.

CMYK Measurement through RGB+I Color Separation

There are quite a number of patents that describe using a fourth channel—an infrared channel— to help differentiate black ink from the others (for example, Frommer, 1968; Keller et al., 1987; Six, 1996; Wang, 1998; Wang and Nemeth, 1999; Ammeter et al., 1999; Ott and Rüegg, 2000).

The IR channel is useful because black ink absorbs IR, while the other three inks do not. Thus, the IR channel image is an image of the black ink by itself. Once we know the amount of black ink, the effect of the black ink can be removed from the other channels so as to determine the amount of cyan, magenta, and yellow.

After the RGB+I values are used to derive CMYK quantities at each pixel, the method proceeds just as the previous method. The cyan, magenta, yellow, and black values are summed over the image to get a measure of how much of each of the inks is present. This sum represents an average ink level over the image.

The RGB+I camera is practical from an engineering standpoint, since it is not difficult to produce IR light. Also, standard sensors (CCD and CMOS) have very good sensitivity in the IR. One engineering drawback is that the cameras are more expensive. Beyond the cost of adding another sensor, RGB+I cameras are not readily available. RGB+I cameras do not benefit from the economies of scale that make RGB cameras inexpensive. Still, the additional cost in is worthwhile since it allows for a system that actually works!

Drawbacks to the CMYK Measurement Paradigm

The idea of CMYK measurement comes from a perfectly reasonable premise. Since ink keys control the amount of ink applied to the web, it is quite reasonable to consider using some measure of the amount of each ink as a means for control. There are also some drawbacks to this paradigm, however.

One drawback is caused by the fact that the measurements of "how much of each ink is present in the image" are not in particularly useful units. Because of this, it is difficult to decide ahead of time what the proper amount of each of the inks should be. This makes it difficult for such a CCIW system to determine target values for the initial color ok. Once there has been a color OK, the system can measure an image to determine target values to hold to. However, the system is not useful for makeready.

A second drawback is illustrated by the "cardinal in the blue sky" image shown in Figure 4. In this image, the blue sky has a great deal of cyan ink, with only a highlight of the other inks. The cardinal, on the other hand is predominantly magenta and yellow.

Figure 4.

In theory, the RGB+I to CMYK conversion will report virtually no magenta ink in the blue sky. In practice, however, small errors in measurement and CMYK conversion will likely cause small errors in the conversion. Darkness in the blue sky pixels may be misappropriated to magenta ink instead of cyan. This error may be insignificant for individual pixels, but it can be an issue since there are such a large amount of blue sky pixels compared to the number of cardinal pixels. The problem is that the magenta conversion errors in the blue sky can overwhelm the actual measurement of magenta in the cardinal. Magenta color moves will be made based on the conversion errors rather than based on making the cardinal the correct color.

A third drawback is illustrated by the three images in Figure 5. Press operators understand that grays and flesh tones are difficult to hold on press. They will naturally sacrifice the color of the tortoiseshell butterfly, the lily, and the Corvette because the eye is more tolerant of changes in these colors. A system that only looks at the total amount of ink will not appropriately weight the importance of different areas of color space.

Figure 5.

CIELAB Optimization

Three other patents (Kipphan et al., 1993; and Seymour, 1999; Darel et al., 2000) describe a completely different approach. Rather than attempt to make sure there is the proper amount of ink, these three patents describe systems that make sure that the *color* (i.e., CIELAB values) is correct.

With CIELAB optimization, CIELAB measurements are first made of each pixel. Next, by comparing these measurements against a target color for that pixel, a color error (ΔL*, Δa*, and Δb*, not a ΔE) is determined for each pixel. Given the ink coverage at each pixel, the system knows how the color of each pixel will change with a change in inking levels. The system decides upon the inking level that comes closest to matching color over all the pixels. This will generally require a compromise, but the tradeoffs are weighed against each other based on how the eye perceives the color difference.

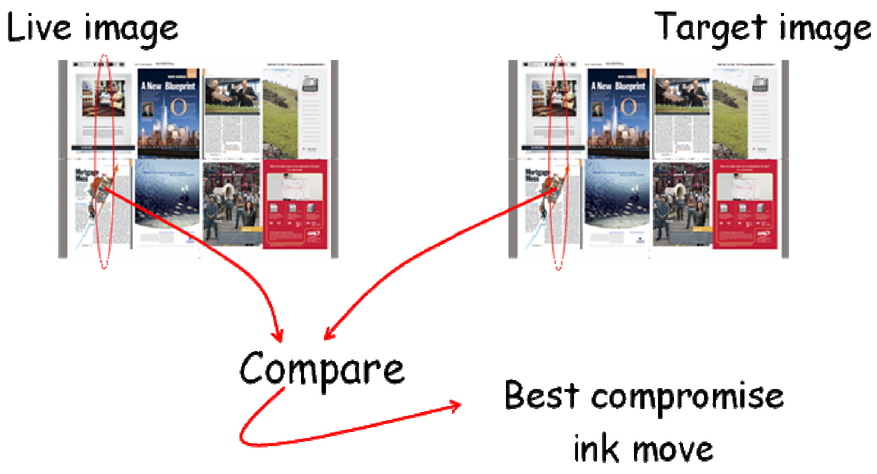


Figure 6.

Creating a target image for CCIW is essentially the same as creating an image for a soft proof. The digital files representing the plates are used to determine coverage values for each of the pixels in the image. An ICC profile is then used to convert from CMYK values to CIELAB.

Inner Workings of the CIELAB Optimization Algorithm

CIELAB optimization will be described here by describing progressively more complicated scenarios.

Case 1 – Solid coverage, single ink.

The simplest case to illustrate is the case of an image which is comprised of only solid coverage of a single ink. In order for the CCIW algorithm to work, it must have some knowledge about how the color of the solid area will react to a change in inking level. This could be done, for example, by printing such a solid, adjusting the ink keys by some known amount and then printing the solid again. This would be the most straightforward approach.

Another approach which is less obvious is built on existing color control systems. There are currently available color control systems that can translate a requisite change in solid ink density to a change in ink key opening. Why not use them? After all, these systems have already solved a lot of the control system problems such as PID loops, adaptive learning of the press, and interface to the press.

It would make sense, then, for a CCIW program to determine suggested changes in terms of changes in solid ink density, rather than in terms of changes in ink key opening. The CCIW program would then pretend to be the solid ink measurement device of a standard color control system.

At this point, this case seems like a ridiculously simple task. It seems that the CCIW program must take measurements of a solid coverage area and then pass those measurements along to the CCS.

The problem is more interesting than it might initially seem, since the CCIW algorithm makes its measurements in CIELAB and the CCS expects its inputs in density. For this simple case, the job of the CCIW algorithm is to make a conversion from CIELAB to density.

Table 1 was adapted from a previous paper by this author (Seymour, 2007 [table 6]). The table shows just such a conversion. As an example, the table says that a one-point increase in solid ink density for cyan (which is a huge change, by the way) will result in roughly a 19.7 point drop in the b* value. These values are called the sensitivity values.

	ΔL*/ΔD	Δa*/ΔD	Δb*/ΔD
C	–17.5	–6.7	–19.7
M	–15.7	19.6	15.9
Y	–4.6	4.9	62.9
K	–27.0	0.1	–0.9

Table 1.

This doesn't quite solve the problem yet, though. We have three criteria to meet. We need to reach the proper values for L*, for a*, and for b*—all three at the same time. This points out the issue that it is not, in general, possible to attain a specific CIELAB value for a given ink. We have one degree of freedom in adjusting inking level, but there are three constraints. As we adjust the inking level, we move along a trajectory in CIELAB space. That trajectory may or may not hit any specific position in CIELAB space.

The solution is to come as close as possible. This is accomplished through the standard mathematical technique called least squares optimization.

Case 2 – CMY gray

In the second case, we look at controlling the color of an area that is a three-color gray. This case works just like Case 1, except that there are now three variables. As with Case 1, it is necessary to know how any ink move will affect the color. There needs to be a table that indicates how a change in each of the three inks (cyan, magenta, and yellow) will change the CIELAB values. The sensitivity values in Table 2 are for illustrative purposes.

	ΔL*/ΔD	Δa*/ΔD	Δb*/ΔD
C	–10	–10	–10
M	–10	+10	0
Y	0	0	10
K	0	0	0

Table 2.

(Note that in Table 2, we see the twelve sensitivity values for a single CMY patch. The three sensitivity values corresponding to black ink are all zero, indicating that adjustment of black ink will not affect the color of this area. Table 1 shows the sensitivity values for four different ink combinations: solid cyan, solid magenta, solid yellow, and solid black. Each of the four solitary inks have twelve sensitivity factors, of course, but most of them are zero. In Table 1, these zero entries were left out so I could make the table smaller.)

With this table, it is possible to write down a set of linear equations that describe the changes in L*, a*, and b* in terms of the changes in cyan, magenta, and yellow. For example, with the values in this table, the change in L* is $\Delta L^* = -10C - 10M$. The change in a* is $\Delta a^* = -10C + 10M$. The change in b* is $\Delta b^* = -10C + 10Y$.

Thus, there are three linear equations, and also three ink adjustments, so the equations can, in principle, be solved exactly. This is at least in theory. The result of the calculation may well yield a ridiculous density. To actually reach a specific color, it may be necessary to increase the density of an ink to, for example, 2.5D or maybe –0.5D. Clearly, in these cases, it is not possible to reach the target color.

Case 3 – Four-color gray

The addition of one more ink changes the set of equations to under-determined. It is now possible to reach any target CIELAB value in multiple ways. How to choose?

Strictly speaking, the standard least squares algorithm is not capable of dealing with this, since it amounts to trying to invert a singular matrix. Singular value decomposition (the Moore-Penrose pseudoinverse) is used to determine a solution.

Case 4 – Two or more areas with different ink combinations and different color requirements

In general, there will be perhaps tens of thousands pixels. Each pixel will have its own CMYK coverage values. Given these coverage values, there are twelve sensitivity values for each pixel.

Each pixel will also have its own color errors. The sensitivity values and color errors will be combined into a large set of linear equations, which will be solved by least squares optimization or by singular value decomposition.

Comparing "CMYK Measurement through RGB+I Color Separation" with "CIELAB Optimization"

CMYK measurement seeks to have the correct amount of ink. CIELAB optimization seeks to have the right color. As such, CMYK measurement is more concerned with the immediate goals of the pressroom. CIELAB optimization is more directly concerned with the overall goal of the process. After all, the customer does not ask for specific number of grams of ink per square meter of paper. The customer asks for a specific color.

These two goals sound very similar. Indeed, under normal conditions, I would argue that meeting one goal will meet the other. But when tradeoffs must be made or when the press is not running under "normal" conditions—which is most of the time—I would argue that it is preferable to make sure that the color of the print is correct.

Since it is generally not be possible to make all colors match the target color, there will always be tradeoffs in CCIW. CIELAB space is preferable to CMYK space or RGB space for making tradeoffs, since it corresponds to the way that we see.

CIELAB optimization deals more appropriately with the "cardinal in the blue sky" problem, since it deals with pixels on an individual basis rather than as an aggregate. The blue sky pixels do not include magenta ink. The algorithm realizes that changing magenta inking levels will not change the color of these pixels.

CIELAB optimization works well in makeready mode. Unlike CMYK measurement, where the initial target values are hard to ascertain, the target values for CIELAB optimization can be derived with an ICC profile. The target CIELAB values at each pixel are as close as one can come to the intent of the customer.

An Engine for CCIW

General Description

Ideally, a camera for CCIW would sample the entire width of the web at 100+ DPI continuously as the web passes by. For each line data point along the line, the camera would collect data from about 30 spectral channels. This data would be acquired every time the web moves $1/100^{th}$ of an inch so at 2500 feet per minute, the camera would acquire 50,000 lines per second.

It would be *possible* to build such a camera with today's technology, at least for a narrow strip of the web, but it would not be economically feasible. We need to scale down the expectations to make the camera commercially viable.

What is feasible today?

Commercially available inspection systems are able to continuously collect images at roughly that resolution and speed, but they are only three-channel devices, instead of the 30-some channels for a typical spectrophotometer. This author has demonstrated that a three-channel spectrophotometer (that is, an RGB camera) is not able to provide CIELAB values that are accurate enough for color measurement (Seymour, 1997 and 2009).

On the other hand, systems used for color control from color bars are capable of acquiring 30 some spectral channels and can sample at well above 100 DPI lateral resolution, but unfortunately they cannot collect data continuously.

What is needed is a compromise—a camera that is economically feasible, but that has enough spectral channels to provide color measurements that are accurate enough, and that has enough resolution so as to provide a useful image.

At QuadTech we developed a camera with such a compromise under the product name AccuCam. In the past few months, the system was installed in two different customer sites.

We first compromised in the number of channels. A three-channel "spectrophotometer" is not quite enough. Standard handheld spectrophotometers sport thirty channels, so thirty is enough. Is there somewhere in between three and thirty that is "accurate enough"? Earlier work (Seymour, 1997) suggested that six channels could be good enough.

Another compromise was to reduce the resolution. While we would have liked (as I said) an image at 100 DPI, our current product will deliver images at 25 DPI. Images collected at 300 DPI, 25 DPI, and 6 DPI are show in Figure 7.

Ozzy Osbourne is
life on the edge —
son Jack's show
e.
Sharon will join him
here celebs perform
uch as skydives and

will appear despite
ath six years ago in an
ash at his Buckingham-

legend, 60 – who once
n stage – needed emer-
breaking his collarbone,
ebra in his neck.
X Factor judge Sharon,
lavish lifestyles behind
wo episodes of the fifth
rne: Adrenaline Junkie,
nmer.

azing

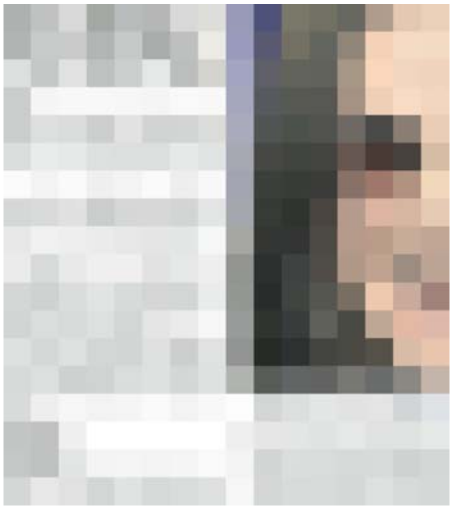
vealed where their jour-
But a source said they

'Bourne to be wild .

four series of MTV hi
new show Osbournes:
But this time daught
coming along for the ri

Ozzy @ 300 dpi *Ozzy @ 25 dpi*

(figure continues next page)



Ozzy @ 6 dpi

Figure 7.

At 25 DPI, it is apparent that the text is no longer readable. The colors in the image, however, are still faithful to the original. When the resolution is dropped to 6 DPI, the image has blurred to the extent that color information has been lost. In our initial research, it was found that CCIW still worked reasonably well at 10 DPI, but that it fell off below that.

The camera operates by positioning itself above the moving web and capturing an image of one entire repeat. After capturing one full repeat of a page width, the camera moves over laterally and collects another full repeat. The system captures a full image of a 64-in. wide web in about 35 seconds.

The collected images are broken down into ink key zones. All pixels in the ink key zones are used to determine the best compromise ink key move. The software has provisions for adjusting the weighting for selected critical areas. For example, ads may be given higher weighting (more votes) than other areas.

Accuracy

Several cameras have been built and tested for accuracy. In the test, the camera scans a coated sheet of 1,296 patches spanning CMYK space. This set includes all combinations of 0%, 20%, 40%, 60%, 80%, and 100% coverage for all four inks. A number of the patches have total area coverage which is beyond the range of normal printing. Being the darkest patches, these present the most difficulty for a measurement device. Rather than set the bar unrealistically high, we chose to eliminate certain of the patches. We decided to exclude any patches with a density above 1.90 D in any of the four Status T channels. As a result, the set of 1,296 patches were reduced to 1,150.

Measurements are converted in CIELAB values and then compared against measurements of the same sheet taken with our reference instrument, which is an XRite DTP 70. The ΔE_{94} values are computed for each patch.

In Figure 8 we see a histogram of the color measurement errors for one typical camera, as measured at the installation site. As can be seen, the results are quite promising. If production tolerances are on the order of 4 ΔE_{94}, then SPC guidelines suggest that a measurement instrument must have an average accuracy of 30% of that, or 1.2 ΔE_{94}. For this camera, the average measurement error was around 0.7 ΔE_{94}.

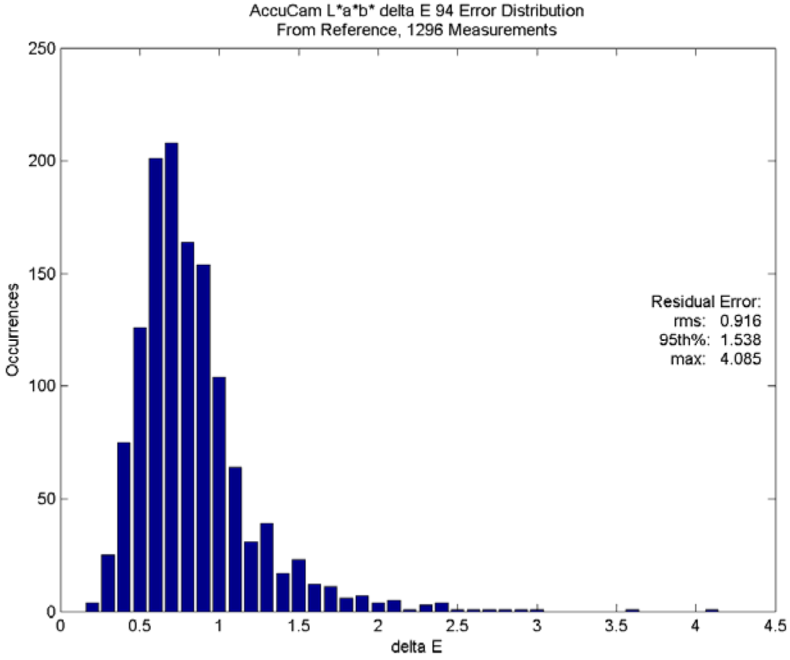


Figure 8.

For comparative purposes, Figure 9 shows the results of a similar test performed using a "high-end" flatbed RGB scanner. (The scanner is referred to as the "Hartford scanner" because it lives at a Quad plant in Hartford, WI.) For this particular scanner, the 95th percentile residual error is 5.40 ΔE_{94}, as compared with 1.54 ΔE_{94} for the six channel camera. Clearly the extra three channels have improved the accuracy immensely—to the point where the camera is useful as a color measurement device.

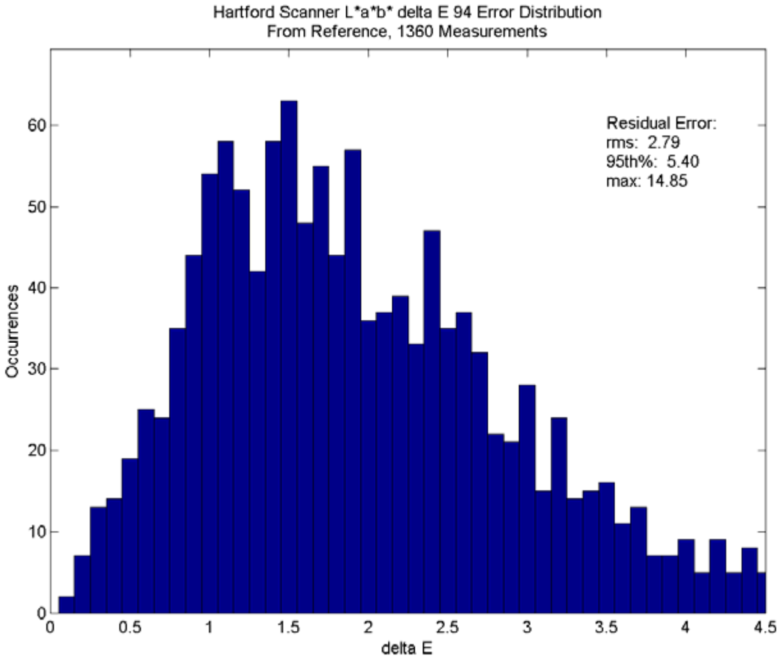


Figure 9.

Results from the Tests

AccuCam has undergone acceptance testing at both customer sites. Since the customers had different acceptance criteria, the tests were somewhat different. I highlight some of the results below.

Tests from Site #1

1. Under control of our system, fine color is achieved after approximately 2,500 revolutions, while in manual mode fine color is generally not achieved. Note: We define "fine color" to be when 80% of all ink key zones in use are within 0.02 absolute delta density of target. This is a threshold intentionally set to a much higher level of quality than that at which color would normally be considered "good."

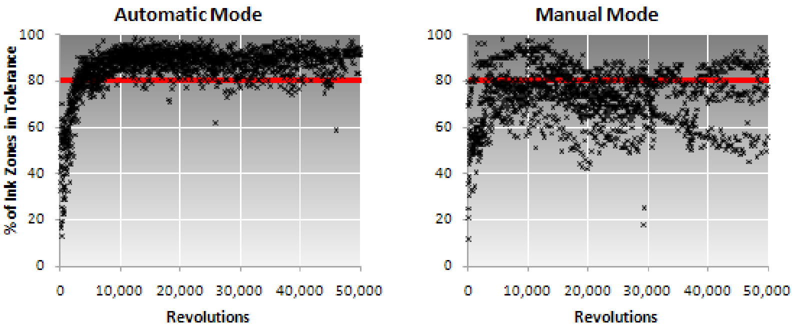


Figure 10.

2. Color quality is held within tighter boundaries under control of our system than under manual control. For example, in automatic mode, our system holds 95% of delta densities within +/–0.03D versus 95% within +/–0.06D in manual mode.

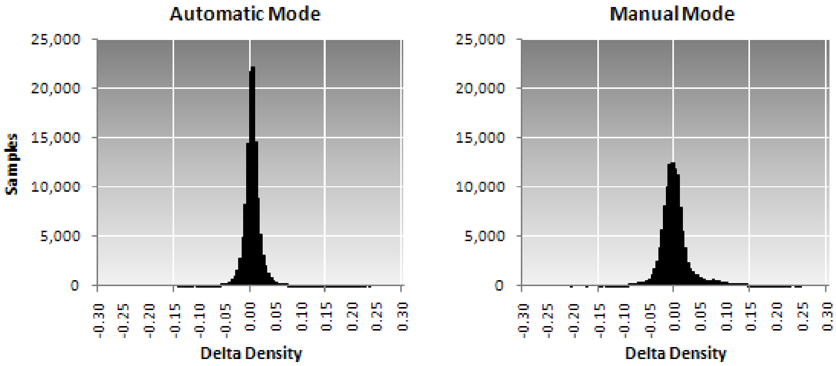


Figure 11.

3. Under control of our system, the first ink key move is issued, on average, after 338 revolutions, whereas under manual control the first ink key moves are sent, on average, after 3,240 revolutions. Even a skilled operator who is completely dedicated to the adjusting color cannot keep up with an automated color control device. Therefore, the color is being corrected much earlier in the run when using our system.

Tests from Site #2

This customer was interested in convergence rate of the average ΔE and in the ultimate accuracy.

In the tests, the ink keys were preset using the normal preset software. The average color error (in ΔE_{00}) was reported as a function of impression count. Figure 12 shows the average over a large number of press runs.

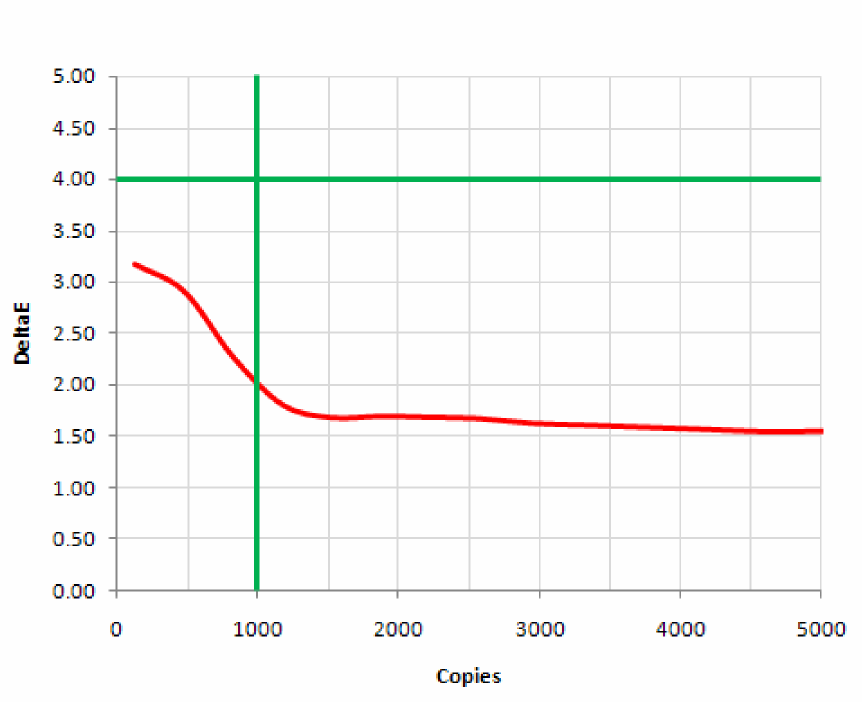


Figure 12.

This test showed that the customer's ink key preset was actually much better than expected—even better than the customer expected. Just the preset alone was good enough to bring the work into the 4 ΔE_{99} color OK. This graph also shows that the system required on the order of 1,000 impressions to converge to an average color error of something over 1.5 ΔE_{00}. This is on par with or perhaps a bit slower than traditional color bar control systems.

Bibliography

Ammeter, Harald, Hans Ott, Nicolaus Pfeiffer, and Manfred Schneider. 1999. U.S. Patent 5,957,049 (Sept. 28, 1999).

Breede, Manfred H. 2007. "The Effect of Ink Film Thickness Variations on Color Control in the Circumferential Printing Cylinder Direction of Offset Presses," *TAGA Proceedings 2007,* p 63–78.

Frommer, Joseph and Warren Rhodes. 1968. U.S. Patent 3,376,426 (April 2, 1968).

Darel, Yair, Miriam Nagler, and Hanan Weisman. 2000. U.S. Patent 6,024,018 (Feb 15, 2000).

Keller, Guido, Andreas Spiess, Hans Ott, and Rold Boegli. 1987. U.S. Patent 4,649,502 (March 10, 1987).

Mason, Robert. 1985. “Specification and control of process color images by direct image measurement,” TAGA 1985, p. 526–545.

———. 1987. “Process control by image color measurement (a progress report),” TAGA 1987, p. 26–40.

Ott, Hans and Kurt Rüegg. 2000. U.S. Patent 6,012,390 (Jan. 11, 2000).

Seymour, John. 1997. “Why Do Color Transforms Work?” *Proc. SPIE* Vol. 3018, p. 156–164.

———. 1999. U.S. Patent 5,967,050, (Oct. 19, 1999).

———. 2007. “How many ΔEs are there in a ΔD?” *TAGA Proceedings 2007,* p. 311–348.

———. 2009. “Color measurement with an RGB camera,” *TAGA Proceedings 2009.*

Sigg, Franz. 2007. “Spatial Uniformity of Offset Printing,” *TAGA Proceedings 2007,* p. 649–658.

Siljander, Roger and Richard S. Fisch. 2001. “Accuracy and Precision in Color Characterization,” *TAGA Proceedings 2001,* p 57–78.

Six, Hans. 1996. U.S. Patent 5,530,656 (June 25, 1996).

Stanford, Alan. 1994 U.S. Patent 5,357,448 Stanford (Oct. 18, 1994).

Van Holten, Erik and Menno Jansen. 2006. U.S. Patent 7,040,232 (May 9, 2006).

Wang, Xin Xin. 1998. U.S. Patent 5,841,955 (Nov. 24, 1998).

Wang, Xin Xin and Robert Nemeth. 1999. U.S. Patent 5,903,712 (May 11, 1999).