

Color Book Production Using T_EX

Michael D. Sofka

Publication Services, Inc.

1802 Duncan Road

Champaign, IL 61821 USA.*

mike@rpi.edu

Abstract

When typesetting a color book the goal is to produce a separate printer plate for each of the colors. The process of splitting the printed output into separate plates is called *color separation*. There are two color separation methods commonly used. *Custom* color separation selects colors from a standard pallet. A different plate is created for each color in the book. *Process* color separation separates the colors into the subtractive color components cyan, magenta and yellow, and creates a separate black plate by a process called undercolor removal. These four plates are used by a printing press to mix the colors on paper when the book is printed. Color separation is a more involved process than simply assigning RGB values to a desktop color printer. This article addresses the issues of professional color separation, and demonstrates how T_EX with a suitable dvi driver can be used to produce quality custom and process color books.

Introduction

There has been recent interest in using color with T_EX. This is evident by macro packages such as Foil-T_EX [Haf92], the discussions about color on NTS-L (New Typesetting System List), and the soon to be standard color support in L^AT_EX 2_ε [GMS94]. This interest was most likely initiated by the availability of low cost desktop color-printers, and the desire to make use of these printers² with T_EX.

Foil-T_EX, L^AT_EX color styles, and other macro packages provide an easy way for the owners of desktop color-printers to use color with T_EX. Their goal is a simple method, using macros and specials, to select color output on a desktop printer. This is different from what is required in color book production. When typesetting a color book the goal is to produce separate plates for each color used by the printing press. An example will help to clarify this. Imagine

that you are typesetting a book that will have red section headings. The final product required by the printer³ is two sets of negative film or camera ready copy.⁴ The first set will be the “black” film, which contains only the black text. The second set will be the “red” film and will contain only the red section headings. But, both sets of film will be printed on a black and white imagesetter because. *it is the responsibility of the printer to provide the correct color ink to the printing press.*

The process of dividing the pages into separate printer plates is called *color separation*. Color separation is a more involved process than assigning RGB values to a desktop color-printer. The question addressed by this article is: “can color separation be done with T_EX?,” and the answer is: “yes, with an appropriate dvi driver.” At publication Services we have been typesetting color books in T_EX since 1987. In 1993 we typeset our first process color book. This was done using a collection of specials that provide information about the current color, its

* The author’s current address and affiliation are Rensselaer Polytechnic Institute, CIS/VCC 312, Troy, NY 12180-3590

² There are unfortunately 3 potential uses of the word “printer” in this article. To avoid confusion, I will use the term *desktop printer* to refer to a low resolution device, color or otherwise, which is used to print files. The term *printer* will refer to a person or corporation that prepares printing press plates and uses those plates to print books. Finally, *imagesetter* will be used to describe a high resolution printer that images to photographic paper or film. An imagesetter is also called a typesetter, but I will use that term for a person who sets type (electronically or otherwise).

³ The person who makes plates and runs the printing press (see footnote 1).

⁴ Film is clear acetate which is used to expose plates for an offset printing press. Film is printed with a negative image, that is transferred once to produce a positive plate. Camera ready copy, or CRC, is photographic film exposed by an imagesetter. Before transferring to a plate, a negative must be made of CRC. Because of the loss of quality when shooting the negative from CRC, negative film is usually requested for color books.

type (process or custom) and its marking model, or how the color interacts with other colors placed on the paper underneath it.

This article will explain the color separation process and describe how \TeX can be used to support electronic color separation. It will concentrate on the PostScript color model, and the quality control steps necessary to insure good color reproduction. Some of the common mistakes made by authors attempting their own color separation will be discussed, so pay attention.

The PostScript Color Model

The PostScript page description language ([Ado90]) has become a *de facto* standard in the publishing world. For this reason I will be discussing color separation assuming the PostScript imaging model. The general principles, however, apply to any color separation, and we have typeset custom color books using Cora ([Lin88]).

An important element of the PostScript imaging model is that all marking is opaque. That is, any mark placed on the page will completely cover (remove) any existing marks it overlaps. This applies equally for solid characters and rules, and for tinted regions. Not all imaging devices work this way, and any attempt to color separate non-PostScript output must take into account the page description languages imaging model.⁵

Adobe defined a series of commands for setting color in PostScript. Some of these commands are now built into level 2 PostScript, but others are conventions defining how to interpret other commands. If you are preparing PostScript files for color separation you should familiarize yourself with these commands and conventions. They are listed in [Ado89], and updated (and simplified) in [Ado90].

Custom versus Process Color Separation. There are two methods of color separation used in book production. These will be referred to here as *custom* color separation and *process* color separation, although the terminology used by other typesetters and by printers may vary. Custom color separation⁶ is the process described in the introductory example. Each element of the book design is set in a specified color. Each of these colors is printed on a separate piece of film. The colors themselves are

assigned based on standard color references similar to those used to select house paints. One common reference for custom colors is the Pantone system ([Pan91a], [Pan91b]). Pantone sells standard approved color charts, and inks. A book design, for example, may ask that Pantone 231 (a light red) be used for all section heads. The typesetter's task in this case is to provide two negatives for each page, the printers job is to prepare plates from these negatives and select an approved Pantone 231 ink for the color plate. If late in the typesetting process the publisher changes the design to use Pantone 292 (light blue) instead of Pantone 231, the printer can supply an Pantone 292 approved ink. The typeset negatives and the prepared plates will not have to be changed. A small sample of Pantone colors can be found on color plate I.33 of [FvDFH90]. If you look up plate I.33 you will note that the actual names are obscured since the colors reproduced in that book are not Pantone colors.

With process color separation each color is separated into cyan, magenta, yellow, which are the subtractive color components. Black is supplied by a process called undercolor removal which removes equal amounts of black from cyan, magenta and yellow.⁷ This is done to provide a better, well registered black. (Imagine how a book would look if all the text was composed of three layers of ink.) The final result is referred to as CMYK color.

Process separation is a more difficult process because of the need to have correctly calibrated colors. With custom colors the printer is responsible for supplying the correct ink. With process colors, on the other hand, the typesetters job is more difficult because correct color balance will depend on the quality of negatives supplied to the printer, as well as the quality of ink provided by the printer. The final typeset output of a process color book is four negatives. Each negative represents one of the cyan, magenta, yellow or black components of the book. The ink will be mixed on the paper by the printing press. Process color is also called 4 color, and the two terms are synonymous in this paper.

In order to provide the correct color mixture the negatives will have screened regions corresponding to page elements. A screen is an area of shading that provides a percentage of the required color. For example, an orange-red can be printed using 0% cyan, 30% yellow, 70% magenta and 0% black. The four negatives must reflect these percentages. In the case of the cyan and black negatives, the color region will

⁵ The imagesetter language Cora, for example, assigns priorities to overlapping tinted regions. The final marks are those of the region with the highest priority. Cora also allows the specification of different tint and pattern values for the intersection of regions.

⁶ Custom colors are often called "spot" colors in desktop publishing and drawing programs.

⁷ This method of undercolor removal can flatten colors and result in too much black on the final print. In practice color balance must be checked and adjusted as needed.

be black because no cyan or black ink is required. The yellow film however, will have a screened area that is approximately 30% filled, and the magenta film will have a screened area approximately 70% filled.

In order to have accurate color output with process colors it is necessary to calibrate your imagesetter to produce the correct screen percentage. Color calibration is notoriously difficult because of the limitations of color monitors, the screen patterns generated by different output devices, and the absorption properties of the paper. Each of these will be addressed below.

Knockout versus Overprint. Another important consideration when separating colors is deciding what happens when one color is placed under another color. The two possibilities are that (1) the first color is removed before the second color is placed. This is called color *knockout*. The second possibility is that both colors print, which is referred to as *overprint*. When designing a book, writing macros, or working on a figure it is important to know which action a color should take. For process separation the default assumption is that the four separations (cyan, magenta, yellow and black) set overprint. This makes perfect sense since the goal is to mix the ink on paper. For custom color separation, however, the default is that custom colors knockout any element set under them. Again, this makes sense when you consider what a custom color represents. A custom color is an industry standard color selected by the designer of the book. Mixing a custom color with any other color will change its appearance from that of the standard.

There is an exception to the general rule that custom colors set knockout. A duotone is a custom color used as a process color. The effect is to mix, for example, Pantone 292 with black to create a variety of colors from pure tone Pantone 292 to pure black. Once again, the intended effect is to mix colors on the paper so the colors must set overprint. In this case, however, the custom color is being used as a process color, and in practice such books are prepared as process color books with 0% yellow and magenta components.

Recall that the PostScript model assumes knockout colors by default. Depending on how your files are separated this could work for, or against you. If the separation is done by manipulating the PostScript color-space (see Rahtz and Goossens this proceedings) then knockout and overprint can be set via PostScript commands. This is the approach Adobe uses in their Separator program. If, on the other

hand, some macro or driver manipulation is being used to remove a color during printing (for example by shifting it off the page) then a page element previously knocked out will now print. This approach effectively sets all colors to overprint. Similarly, if separation is done by simply changing the PostScript `setgray` value then all colors are effectively set to knockout.

Ink Order. When preparing color separated output it is often important to consider the order in which colors will be applied to paper by the printing press. There are three rules to remember: (1) black is set last on the press, and (2) yellow is very light and tends to get lost in darker colors, and (3) black will effectively cover most other colors.⁸

Applying these rules we see that if, for example, yellow text is being set on a solid black box the yellow must knockout the black in order to be visible in the final book. If black text, however, is being printed on a solid yellow box, the black does not need to knockout the yellow. In fact, it is preferable to have the black overprint the yellow to avoid problems with trap (see below).

Technical Difficulties.

The task of color separation is conceptually very simple, but in practice can be fraught with difficulties. Most of these difficulties stem from the process of re-integrating the separated plates. In order to print color books the process of applying ink to paper must be understood. Many of the problems that can occur with this process are under the direct control of the typesetter.

Registration. Once the separations are made it is necessary for the printer to realign them correctly on the paper. This process is called registration, and it is a subtle point that is often missed by authors preparing their first color separated book. In order for the printer to be able to re-align separated output the output must include registration marks. These are alignment marks that are printed on every page regardless of color. By aligning the marks a correct composite should be obtained.

Registration also refers to the quality of alignment in the separated output. Obviously, the registration marks should set in the same position on each page, but less obvious is the effect that image setter capability, and film quality can have on registration. Some imagesetters are rated for color by specifying the repeatability of the output. The

⁸ The press order is usually yellow, followed by cyan, then magenta and black. If an additional layer is being applied it may be set earlier or later depending on the design.

repeatability is usually specified as the difference between negatives (in mills), and the time frame over which the repeatability holds. That is, if a negative is printed on the 1st of the month, how likely is it that the registration will be within 1/10 mill if the negative is reprinted at the end of the month. Registration is also affected by the weather. This is because the acetate used for negatives will stretch or shrink slightly as the humidity changes. Maintaining a constant work environment is therefore important for good registration, and some imagesetters include humidity and temperature control units to maintain repeatability.

Tagline. To keep track of each color of each page the printed negatives should all contain identifying information. This is called the tagline, and it can go a long way towards avoiding confusion. The tagline is less important for one color books since most pages will have a folio (page number). Color books, however, will have one or more colors printing without a folio. Some pages, in fact, may be completely blank except for the tagline and registration.⁹

Unless you have actually tried to keep track of several hundred pages of negatives (in a shop that is printing more than one book) it is difficult to appreciate how important taglines are. The usual procedure followed by printers if they receive a page without a tagline is to return it. A tagline should include at least the folio and color. Some identifying name or title will help avoid confusion with other books.

Trap. Registration is only one aspect of realigning separated plates. Since the output of \TeX is specified in scaled points, and imagesetters print at 2540 dots per inch or more, a high degree of registration is possible. But, there are cases where exact registration is not wanted. When setting elements in different colors next to each other (elements that will separate to different plates) it is necessary to provide a small region of overlap to prevent the white paper from showing through. This area of overlap is called trap. Trap is usually specified in mills, with 3 mills being a typical value. It is not a large area of overlap, but it is important.

An example requiring trap can be seen in figure 1a. The figure sets a region of Pantone 231 (red) over a region of black. Even the most exact press alignment is likely to allow a small gap of white paper to show through. To prevent this, the Pantone 231 must be trapped against the black.

⁹ Blank pages are required to build the imposition. The printer has no idea what belongs on each page of an imposition and missing pages can cause confusion, delays and errors.

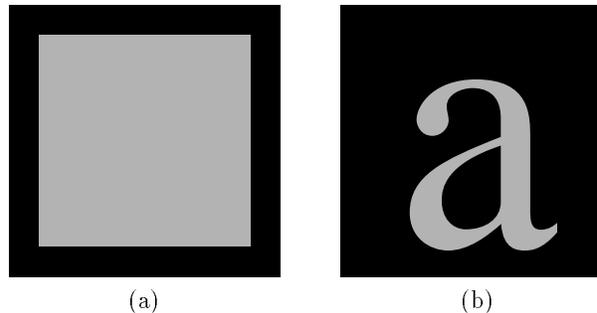


Figure 1: Examples of trap. (a) Red screen over black. (b) Red character over black requires knockout and character trapping code.

Note that the red square cannot just overprint the black. Instead, what is required is that the black be knocked out, and then the red set overprint overlapping the black by 3 mills. The ability to set knockout *and* overprint make trap possible. You could argue that if all colors set overprint, the black frame in figure 1a can be set with rules. Then a single red rule can be set inside the black frame. This will not work for figure 1b, which requires special support for trapping font characters.

Screens. Perhaps the single most important technical difficulty with color separation is screens. A screen is a area of the paper that is not 100% filled with ink. The term tints is often used, and you can observe the effect of screening by using the PostScript `setgray` command. When setting a photograph the screen is called a halftone. Screens are used to create a region where color is not fully saturated. This is done by setting a pattern of dots which partially fill the region. The reason for using dots is that desktop printers, imagesetters and presses either place a dot, or they do not. There is no mixture of, for example, cyan and white ink to dilute the color.

While setting a screen in PostScript is simple in principle, in practice it requires special equipment and attention to detail. Desktop printers are incapable of setting correctly screened output. This is a limitation of their marking engines—300 or 600 dots per inch is just not enough dots to provide good screens. In addition, the paper and toner used by a desktop printer have too much dot-gain (see below) to present an accurate screen.

PostScript screens are a printing device dependent feature. That is, the method by which a screen is set is not defined in the PostScript language. This was most likely an intentional design decision since the best method of setting a screen will depend on

the qualities of the marking engine and print media. In addition, screen generation is an active area of research, and each imagesetter company has its own brand of screening technology for sale. Screen type and quality is a very important consideration when purchasing an imagesetter. Perhaps *the* most important consideration when printing separations for a process color book, because bad screens equals bad process color.

Screen Attributes and Types. Different imagesetters are sold with different screen types installed. Before discussing the types of screens available we need to understand how a screen works. In PostScript the `setscreen` command is used to change default screen qualities. It takes three arguments: frequency, angle and procedure. The frequency is how many lines per inch are represented in the screen. For example, in figure 2a the two screens differ only in frequency. The angle is the angle of the lines composing the screen. The screens in figure 2b differ only in angle.¹⁰ Finally, the procedure defines a function to determine the order in which halftone cells are filled to produce the desired shade of gray.

Not all combinations of frequency and angle can be set by an imagesetter. The actual values supported vary with screen type (the procedure) and the resolution of the marking engine. The best output is usually obtained for only a subset of the possible values. In addition, the printer may request a particular range of frequency for best pickup while making the press plates since too high a frequency may not transfer, or may result in dot-gain. The best advice to offer an author preparing color separated output is to not use the `setscreen` command unless you know what type of imagesetter will be used to print the film, and only after conferring with the operator of that imagesetter.

Most high resolution PostScript imagesetters are capable of *Rational Tangent* (RT) screens. These screens are limited to angles which have tangents that can be represented as a ratio of two integers, and frequencies that evenly divide the device resolution. On imagesetters sold for color work it is common to find *Irrational Tangent* (IRT) screens. As the name implies, IRT screens can represent angles whose tangent is a real. IRT screens are a minimum for process color books, but they suffer from limitations which result in a poor reproduction. All imagesetters sold for 4 color books have some, usually proprietary, screen system included. The latest screens

¹⁰ Because of PostScript device limitations the screens of figure 2b may differ in angle and frequency. The actual PostScript command requested only that the angle be changed. This is a good example of the device dependency of screens.

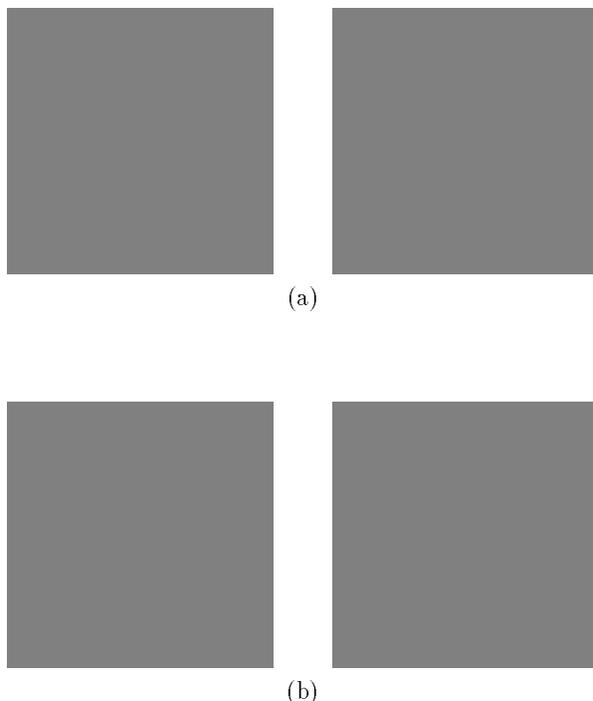


Figure 2: Examples of the effects of frequency and angle on screens. (a) Screens differ only in frequency. (b) Screens differ only in angle. N.B. The screens you see may vary with the display device and PostScript interpreter.

employ stochastic methods to eliminate repetitive patterns which result in moiré. These methods are usually built into the PostScript interpreter instead of defining a PostScript procedure with `setscreen`. This is a matter of efficiency since screen calculation can consume the majority of the CPU cycles in an interpreter.

Screens in Process Colors. If you are typesetting a process color book, and intend to print your own output then you will have to buy screens. There are many screens available, and most of them come with an imagesetter. This is only a slight exaggeration because screen type and quality are dependent on marking engine ability. Some companies do, however, offer screen updates to existing imagesetters. These work by changing the PostScript procedure, and can be slow as a result.

The reason that screens are so important for 4 color books is that process color separation requires that screen areas overprint each other. When this is done, a variety of undesirable side-effects may occur. The most common problem is moiré. In traditional color separation the halftone screens for

each of the colors are rotated so that the dots overlap to form circles called rosettes ([Agf90], [Bru86]). The screen angles used are 105°, 75°, 90° and 45° for cyan, magenta, yellow and black respectively ([Agf90]). The rosettes formed are invisible to the unaided eye, and instead the illusion of color is created. If the dots are not placed accurately, however, moiré will result. Moiré is caused by an interference pattern between the screens destroying the illusion of color. Most PostScript printers (and this includes a large number of imagesetters) are not capable of placing dots accurately enough to avoid moiré. Even with a color ready imagesetter, special screening methods are required since there are just not enough pixels to create an accurate circle at the size required. Some of the methods used are:

- Oval, or other non-circular patterns which overlap to form larger rosettes.
- Randomized noise added to the patterns to disturb the regular interference which causes moiré.
- Micro-dots (very small dot patterns) that are too small to overlap into rosettes. Micro dots can be set in a random pattern to create stochastic screens.
- Very high resolution output. It is not unusual, for example, to find advertisements for 5300 dot per inch imagesetters. These imagesetters are more capable of holding a “hard” dot (an industry term for identically shaped round dots) in screen patterns.

Each of the above has its advantages and disadvantages. The most common disadvantage is that variations on screening methods can be very slow—especially when random noise is being added. Stochastic screens usually suffer from larger scale patterns caused by poor randomization algorithms or small cell size. The result is, once again, moiré.

Other Technical Difficulties. Moiré is not the only problem encountered with screens. The following list of difficulties must all be overcome in some way by the imagesetter and printer. The compositor's responsibility is to provide the best possible output that reduces problems for the printer.

- Dot gain.

Dot gain is an increase in the size of a halftone dot from the time the negative is printed by the imagesetter until the final paper is printed by the press ([Gre93]). Dot gain will affect the amount of ink transferred, from the plate to the final paper. The factors affecting dot gain are paper absorption, screen frequency (higher frequency = higher dot gain), and ink thickness.

You can think of dot gain as the amount smear that takes place on the final printed book.

- Processing Speed.

It has already been mentioned that some screening methods take a lot of time to process. It is not unusual for a simple switch from IRT screens to a randomized oval screen to increase processing time by 800% or more. When the IRT screened page took 15 to 20 minutes to print the slowdown for oval screens can be significant.

- Screen Models and Patents.

Many screening methods have been developed by printing and imagesetter companies and are covered by patents. These are not necessarily software patents since the original patent was granted when a “screen” meant a physical piece of acetate which is laid over a color photo when shooting negatives. The extension of the patent to software embedded in a RIP is more natural than in many other software patents. A company using patented methods without paying a license fee, or purchasing equipment and programs from the imagesetter manufacturer would be infringing.

Screens and Custom Colors. While screen quality is important with custom color books it is not as critical a component as with process colors. Usually RT screens provide output of sufficient quality. This is because process colors do not mix (unless doing a duotone), and both black, and custom colors are printed with 45° screens (whose tangent is 1/1). As a result, screens do not overlap and interfere with each other. It is important, however, to be able to provide screens with a dot size (frequency) that can be photographically reproduced when making press plates. It is also important that the screen density match the specifications, and that the density be even within and between the screened areas. This is usually beyond the capabilities of desktop printers, and publishers will typically avoid screens when authors are preparing camera ready copy.

Color Calibration. When preparing and testing process color separated negatives it is important to maintain correct color balance. A variety of conditions can affect the final appearance of your output. The most basic is the calibration of the imagesetter. When a 20% screen is requested the output should be a screen that provides 20% fill of the area after printing. If you are proofing the color then all proofing devices from monitors to printers and photographic based proofing systems must be calibrated to provide the best color fidelity they are

capable of. Final decisions about color should not be made on the basis of desktop color printer output. Finally, the conditions under which colors are checked must be constant to avoid metameric color matches ([Bru86]), or other light dependent color changes. This may require building a color proofing room with controlled lighting.

CMYK color space is based on the absorption properties of particular inks, but ink batches can vary from printer to printer. For this reason CIE color-space is often suggested for internal calibration. Then standard printer samples (available from film suppliers) can be prepared showing a match between the local color calibration and final product. If the final color is incorrect, it can then be shown to be a problem with the plates or printing press, and not a problem with the negatives.

Color Proofs. It is important to check color in house before printing negatives, and to test those negatives before printing the book. When labor is considered, a single page of negative film can cost \$10 (US) or more, and a page of imposed film will cost even more. If the final printing begins without a less expensive color check (or with no color check), schedules and budgets may slip, and unlucky compositors could find themselves financially responsible for a bad press run. Color proofs can also serve a contractual purpose.¹¹ For example, a page may be approved for final film based on the output of a desktop color printer, or some other color proofing method. If this is the case the client and typesetter must agree on the method to be used, and they must be aware of the limitations of the chosen method.

The only real test of color fidelity is the final printed book. This is because the actual colors will depend on the ink batch, paper, and press calibration. It can be expensive, however, to setup a press run for color testing. A variety of methods are available, therefore, to check color before the press. In general, there are two broad classes of color proofing systems—those that test the PostScript files, and those that test the negatives. Each has its place in that it is less expensive to catch errors before printing film, and before using that film to print a book. PostScript files can be checked using a color monitors or color desktop printer. Depending on the type of color being used, negatives, can be tested using blues (ultraviolet sensitive paper exposed through negatives), cromalin (dry power colors), photographic paper exposures, or other systems. Each of these color proofing methods have

their own tradeoff in cost, time to prepare proofs, and the quality of the proofs.

There are two reasons for color proofing. The first is for color breaks. That is, testing that each element is in the correct color. For custom colors this is usually all that is required, and desktop color printer output is often accepted as proof that the file's colors are correct. Likewise, blues or visual inspection can be used to check that the that custom colors were separated correctly to the final negatives. Grayscale desktop printer output can also be used if each color is printed at a different tint percentage, and separated grayscale output can be used to check the separations. It usually requires some negotiation, however, for clients to accept grayscale output as a color proof.

Process colors must be tested for both color breaks, color fidelity, and moiré. This usually requires that the negatives be tested using cromalin or photographic processes. There are, however, desktop color proofing systems that send separated files to a color desktop printer. These systems allow for moiré to be checked before printing negatives, and provide near photographic color output for checking fidelity. All of these systems, however well calibrated, do not provide perfect proofs. Once again, the only true test of final color is a press run using the same quality materials that will be used in the book.

Using T_EX

After considering the above the obvious question is: “can T_EX be used for professional color book production?” The answer is yes, but it requires dvi driver support, and the typographer should be aware of the procedures involved in plate making and book printing. First, be aware that most imagesetter manufacturers, and consumable suppliers will not know about T_EX so you can expect little technical support when printing. On the other hand, most of the actual problems with color calibration, moiré and dot gain are common to all color separation electronic or manual, with or without T_EX. If you buy an imagesetter you are also buying expertise in its use. Use your purchases to leverage help. Second, many of the issues of color fidelity and moiré apply more often to photographs than the typical T_EX element. Solid color, however, is not immune to moiré and the effect looks very bad in printed books. In addition, the colors used in T_EX may need to match those used in figures, and small differences in the screens and density can produce a noticeable difference in the final product. Finally, color books, especially process color books, tend to have more

¹¹ Thank you to an anonymous reviewer for pointing out this fine point of client typographer interaction.

complex designs where text and figure elements interact with each other. A figure, for example, may be surrounded by a color box or head-element that must trap with the figure. So all of these checks and balances must work with \TeX as well as with figures and photographs.

\TeX provides a powerful macro language that can make the process of managing color elements very easy compared to the more common desktop systems. Adding color late in a book design, for example, can usually be accomplished with a macro change. Changes to the selected color can likewise be affected by changing macros. Low resolution “placement only” figures can be included by using a \TeX conditional, or via driver switches. In short, we have found that using color with \TeX is not only possible, but that \TeX helps the process by virtue of being programmable.

Specials for color. Color separation in \TeX requires dvi driver support. The specials for color separation must convey the following information.

- The CMYK color values.
- process versus custom color separation.
- knockout versus overprint marking.
- foreground percentage.

The first requirement is obvious. The values of cyan, magenta, yellow and black are basic to determining the final color for process separation, and for proofing both process and custom colors. Process versus custom determines the type of separation to be used for a color, and knockout versus overprint is for specifying what happens to colors set below a new item. Note that black is a color, and the specials must keep track of when black is knockout, and when black is overprint.

Specification and interpretation of the CMYK values, however, change with the color model. When a process color is being set the CMYK values represent the actual percent values of cyan, magenta, yellow and black ink required on the paper. A change to these values will represent a change to the final color. For a custom color, however, the CMYK values represent an approximation to the final color to be used by proofing devices, and it in no way affect the final book color. Again, this is because it is the printer’s job to supply the correct custom ink for the press.

The above is an important point that is central to doing color separation correctly. When specifying a color the goal is to provide the best possible quality in the final printed book. But, the values of CMYK which produce the best book color are unlikely to produce the best desktop printer or mon-

itor colors. When printing custom colors this is of no consequence, but when printing process colors it has a number of consequences that can affect the final product. Process color separation shifts the responsibility for good color from the printer to the typesetter. A good color special, therefore, may provide different values of CMYK for final separation versus composite proofing. In fact, different values of CMYK may be required for each output device that might be used to proof the pages.

One approach to selecting the correct custom color CMYK values is use a standard set of color names based on the Pantone color charts. The special would then specify a color name which keys the correct CMYK values. A different CMYK value would be required for each printing device. For process colors the task is slightly more difficult since there generally are no standard names. A local file which provided generic names such as “light-green”, with the correct CMYK values for each device can be used instead. Alternatively, the special can store a set of CMYK values, with one value designated the “correct” value.

Some of you may think: why not use \TeX to define a conditional which selects the color value. We have done this, but the problem we encounter is that the resulting dvi file is device dependent, and would have to be re- \TeX ed with the correct flags in order to produce correct printed output. This has resulted in a lot of lost time and material, and we avoid such device dependencies whenever possible. When they cannot be avoided, a “printertype” special is used to tell the driver which printer the dvi file is prepared for. A mismatch in printertype aborts printing. This is slightly inconvenient, but much less inconvenient than printing 100 pages of incorrect output (at 1-20 dollars per page depending on output device).

The default value of knockout and overprint differ for process and custom colors. This is because custom colors are defined based on standard color charts and inks so it is undesirable to mix them on paper with any other color. Process separations, however, are supposed to mix on the press. This does not mean, however, that custom colors should always knockout and process colors should always overprint. When trapping with a custom color, for example, it is necessary to overprint. Likewise, it may be more convenient to knockout a section with a process color (which must then be separated into overprinting CMYK values), then to typeset around the knockout (see figure 1a). A way is needed, therefore, to change the default behavior of a color.

When typesetting a color book the goal is to produce a separate printer plate for each of the colors. The process of splitting the printed output into separate plates is called color separation. There are two color separation methods commonly used. Custom color separation selects colors from standard pallets. . . .

Figure 3: Back text set on a red 30% screen. Since black prints last, and is dark it can overprint the screen without trap.

Finally, the foreground percentage determines if a color is set at 100% of its stated value, or at some lesser percentage. A custom color may be set at 100% value which means that the negative will be clear where the Custom color should set. Just as often, however, the custom color may be set at some percentage of its value. For example, in figure 3 the black text is set over a 30% screen. The screen is a custom color (and the black will overprint so that trap is not necessary). When setting a custom color at a reduced percentage a new color cannot be used since it would then separate onto a different negative. That is, if we are setting Pantone 231 at 30% we want both the solid 100% Pantone 231 and the 30% Pantone 231 to separate to the same negative. We need a way, therefore, to change the foreground value at which any rule, screen or character will set.

Process colors can also be set at a reduced percentage. This does not affect the separations since all process colors will be split into CMYK components. The changed foreground, however, will change the resulting CMYK values, and the method used should provide good output. Often a direct percent reduction will suffice, but special settings may be required depending on how the proofs look. A red, for example, may appear too pink at a reduced percentage and a new color balance may be chosen.

The Specials. In the dvi driver `dvips82` used at Publication Services we make use of the following specials to define and use colors. All of the specials follow a simple syntax of `<name> <arguments>`, where the `<arguments>` are `<key>=<value>` pairs. Standard commands which accept any T_EX unit of measurement are used to read dimensions from specials. These commands convert the values into scaled points for internal processing.

DefineColor. The `DefineColor` special is used to provide an internal symbolic name for a color. Define color establishes the CMYK values, model and

overprint value. It can also define a tint value to be used when printing on a gray-scale desktop printer. The format of the command is:

```
DefineColor
  <name>=[color(<c>,<m>,<y>,<k>)
          |pms(<pantone-name>)]
          |alias(<color-name>)]
  [separation=[process|custom]]
  [overprint=[true|false]]
  [tintpercent=<%>]
```

`<name>` is the symbolic name which is used to refer to the color from this point on. The name is set to one of three definition types. The first provides the cyan, magenta, yellow and black values for the color, while the second provides a standard Pantone name which is looked up by printer type. The last definition defines the color as an alias of a previously defined color. All aliases of a color will separate together with that color, but they can have different separation, overprint and tintpercent values.

The value of `separation` can be either `process` or `custom`, with the latter being the default. The `overprint` argument defines the color as either overprint (`overprint=true`), or knockout (`overprint=false`). If the separation is process, then overprint defaults to `true`, otherwise it defaults to `false`.

The value of `tintpercent` is used for checking color breaks on a gray-scale desktop printer. It is not possible to see colors on a gray-scale printer, but worse the colors may be metameric in gray. That is, even though they are distinct colors, they appear the same in black & white. In order to aid proofing a `tintpercent` value can be specified. When printing a composite to a gray-scale printer the tintpercent will be used for all page elements in that color. With the `alias` color definitions different values of `tintpercent` can be used with, for example, knockout versus overprint versions of the same color.

SetColor. Once a color is defined it can be used to change the state of the current color. This is done with the `SetColor` special, which is defined as:

```
SetColor color=<name>
```

where `<name>` is the symbolic name of a previously defined color. All rules, characters, screens and figures from this point on will be set using the attributes defined for `<name>`.

KnockOut, OverPrint and DefKnockOut. The use of aliased colors allows the definition of knockout and overprint versions of the same color. The knockout and overprint values, however, can also be changed using specials designed for that purpose. The special `KnockOut` sets the global color state

to knockout. All colors, regardless of their definition, will now set knockout. Similarly `OverPrint` sets the global color state to overprint. The special `DefKnockOut` restores the color state to that specified in `DefineColor`. There are also versions of these specials for setting knockout or overprint for the next single rule, character, screen or box encountered.

SetForeground. The current foreground percent can be changed with the special:

`SetForeground fg=<n>`

where `<n>` is the desired tint value.¹² The default value is 100%, and any value between 0% (printing white) and 100% (printing full color) is allowed. Changing the foreground percent does not change the current color. Instead, all rules and characters are set as a `<n>%` screen.

Trapped and Abutted. Trapping control is supplied by the macros:

`Trapped trap=<m.n>`

`Abutted`

where `<m.n>` is some dimension. When the value of `trap` is non-zero, all rules, characters and screens are set trapped by the trap amount. This is accomplished by first setting the element in knockout, and then setting it a second time overprint. The overprint is stroked by twice the trap amount. The effect is a region of overlap between the trapped element and anything it prints over. `Abutted` is the same as “`Trapped trap=0pt`”, and there are also `TrapNext` and `TrapBox` specials to trap only the next element or box.

ScreenR, TintRule, TintChar. In addition to commands to define and change color, overprint and trapping a color book can benefit from commands to set screened elements. This can be accomplished via PostScript, but it is more efficient and easier to have builtin commands. The `dvips82` driver has a variety of commands to set screened areas (with and without rounded corners), circles and characters. Each of these commands allows a trapping specification as well as a background percentage. `dvips82` places all screens defined with these specials under all characters and rules. The specials are based on the tints and patterns commands available under Cora on an L300 imagesetter.

Macro Considerations. The macros that use these specials are for the most part simple and strait forward. But, there are special considerations around page boundaries. Earlier versions of `dvips82` did

¹² If I were to design the system today I would spell foreground correctly.

not keep track of the color state in effect at the beginning of each page. If a macro restores color after, for example, a box then the restoration of color could appear on the next page. This caused problems when printing selected pages, or when printing to a face up device. The macro fix was to have the output routine save and restore all order dependent states with each page break. Later versions of `dvips82` perform a preprocessing pass on the dvi file. This allowed optimization of font memory, and the recording of color, foreground, overprint, and so on with each page. Pages selected or rearranged (for example, for imposition printing) still cause problems, however, so that the macro approach has been maintained.

This leaves only the problem of color changes within paragraphs. If the color changes in a paragraph, and changes back before the end, and if a page break occurs in the color section, then the macro based color state will be incorrect at the top of the page. The only `TEX` mechanism for handling this would be marks. This type of design, however, is so infrequent that we have rarely had problems of this nature.

The Separation Process. There are two separation methods supported by `dvips82`. The first method creates a *separator compatible* PostScript file. This is a file that follows the Adobe conventions for color PostScript files, and which can be color separated in a manner similar to that used by Adobe Separator. Adobe Separator color separates a file by including PostScript commands that redefine `setcmykcolor`. A special PostScript command is used for custom colors so that they can be distinguished from process colors. It also allows a custom color to be converted into a process color, and separated into its CMYK components. At Publication Services we have written a UNIX version of separator that works with Adobe Illustrator and `dvips82` files.

The advantage of this method is that color art can be integrated into the PostScript file before separation, allowing art and text to be separated together. For this to work correctly the symbolic *custom* color name defined with `DefineColor` must match the Custom color names defined in figures. If the names do not match then, as far as separator is concerned, they represent different colors. This can be a problem when using art prepared out of house by the author or another supplier. Hence, UNIX separator (and Adobe Separator) allow different colors to be combined onto the same negative.

The second method uses the driver to do all color separation. This was a very easy addition to `dvips82` since all pages were handled in two passes. The first pass stores all rules, characters and other page elements in a table, which the second pass prints. This was originally done in order to place all screens below text (mimicking Cora), but it also made electronic color separation easy. If an item is in the current color it is placed in the page table. Otherwise, it is left out. Each element stored in the table has an associate set of attributes including its current color. Process color elements are printed tinted according to the value of cyan, magenta, yellow or black (depending on the requested separation). Knockout is handled by setting knockout colors with a foreground of 0

`dvips82` also contains a color proofing mode which places all elements on the page table, but tints those not in the selected color. This is very helpful for checking color breaks (confirming that each element is in the correct color). The tint value is the `tintpercent` defined with the color, or 75% if no `tintpercent` was specified.

The disadvantage of the driver based method is that only color separated figures can be integrated. As a result the figures must be pre-separated. Future modifications include having `dvips82` run the separation program on figures. For this to work all figures must use consistent color names, which is, once again, a problem for art prepared off-site.

Summary and Conclusions.

Professional custom and process color separation can be done with T_EX and the right set of specials. The specials listed above are what we use with `dvips82`. They encompass some qualities that I have not seen in other color specials such as color aliasing, and support for gray-scale proofing. The use of knockout and overprint colors is needed in order to be able to trap correctly.

There are a variety of desktop systems that support color, so one may ask why we do not use them. The answer is that we do when they are the right tool for the job. Very often T_EX is the right tool for the job, and T_EX can easily be extended through specials to equal and exceed the color separation abilities of any desktop system.

Finally, I am very encouraged by the work being done with `dvips` and color. By adding support for professional color separation to `dvips` the task of converting an author's L^AT_EX files into professional quality negatives will be made much easier.

Bibliography

References

- [Ado89] Adobe Systems Incorporated. Proposal for color separation conventions for postscript language programs. Technical Report 5044, Adobe Systems Incorporated, December 1989.
- [Ado90] Adobe Systems Incorporated. *PostScript Language Reference Manual*. Addison-Wesley, Reading, MA, second edition, 1990.
- [Agf90] Agfa Compugraphic Division. *An Introduction to Digital Color Prepress*. Agfa Corporation, 200 Ballardvale Street, Wilmington, MA 01887, 1990. Descriptions of color models, trapping, halftone and screens.
- [Bru86] Michael H. Bruno. *Principles of Color Proofing: A manual on the measurement and control of tone and color reproduction*. Gama Communications, P.O. Box 170, Salem, NH 03079, 1986.
- [FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, New York, 1990.
- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, New York, 1994.
- [Gre93] Gretag Color Control. *Applied Densitometry*, second edition, 1993.
- [Haf92] James L. Hafner. FoilT_EX, a L^AT_EX-like system for typesetting foils. *TUGboat*, 13(3):347–356, October 1992.
- [Lin88] Linotype. *Linotronic 300/500 Image-setter Interface Manual*, 1988.
- [Pan91a] Pantone, Inc. *PANTONETM: Color Specifier 1000/Coated*. Pantone, Inc., 55 Knickerbocker Road, Moonachie, NJ 07074-9988, 1991.
- [Pan91b] Pantone, Inc. *PANTONETM: Color Specifier 1000/Uncoated*. Pantone, Inc., 55 Knickerbocker Road, Moonachie, NJ 07074-9988, 1991.