`DVISep` — A colour separator for DVI files

Angus Duggan

Harlequin Ltd., Barrington Hall, Barrington, Cambridge CB2 5RG, United Kingdom
`angus@harlequin.co.uk`

Introduction

`DVISep` is a simple colour separation program for
DVI files. It reads a single DVI file, outputting
separation files for each process colour and spot
colour found in the input file. `DVISep` recognises
the colour `\special` commands used by Tomas Ro-
kicki's `dvips` driver, but is not limited to use with
`dvips`. `DVISep` is part of the second release of
the author's DVIUtils package of DVI manipulation
programs.

Colour printing presses and printers normally
use four *process* colours (cyan, magenta, yellow and
black) printed on top of each other to create the
illusion of many colours. Cyan ink absorbs red
light, reflecting only blue and green. Magenta ink
absorbs green light, reflecting red and blue, and
yellow ink absorbs blue light, reflecting red and
green. Black is also used as a process colour because
the cyan, magenta, and yellow inks do not have
perfect absorbtion, and a combination of solid cyan,
magenta and yellow usually looks a muddy brown
colour instead of truly black.

Colour separation is the process of splitting the
image into appropriate proportions of the process
colours for the colours desired. `DVISep` generates
separate files for each of the process colours, contain-
ing only the parts of the image with the appropriate
colours in them.

In addition to process colours, printed pages
may use *spot* colours. Spot colours are used in
several circumstances; when there are only a couple
of colours in a document, it may be cheaper to
print it with spot colours rather process colours;
special inks (*e.g.* textured, metallic, neon colours
or luminescent colour) are required sometimes; and
sometimes it is necessary to provide an exact colour
match, for instance in a paint or ink catalogue.

Having decided which colours to use, there are
still different ways of combining those colours on
the page. Each object on the page may *knockout*
or *overprint* other objects. Knockouts cause blank
areas to appear on other separations; this may be
useful when an ink should not be combined with
other ink on the page; for instance, a spot colour

should probably not be printed on top of a process
colour background.

Figures 1 and 2 illustrate the difference between
overprinting and knockout. If knockouts are set, the
shapes drawn last erase the corresponding areas of
shapes drawn before them; if overprinting is used,
the colours of previously drawn shapes are combined
with the latterly drawn shapes.

Using `DVISep`

The simplest use of `DVISep` takes an input filename:

```
dvisep file.dvi
```

This generates the output files `Black.dvi`,
`Cyan.dvi`, `Magenta.dvi`, `Yellow.dvi`, and
spot`.dvi` for each spot colour used in the input
file.[1]

The `\special` commands currently recognised
by `DVISep` are a subset of those defined by Tomas
Rokicki's `dvips` driver. The `\special` commands
start with either `color` or `background`, followed
by a colour specification. The colour specification
may be either a colour name, for example `Maroon`,
or the name of a colour space (`rgb`, `hsb`, `cmyk`, or
`gray`) followed by an appropriate number of numeric
parameters (3 for `rgb` and `hsb`, 4 for `cmyk`, or 1 for
`gray`). The numeric parameters are all within the
range 0–1, indicating the intensity of each colour
component. Note that for additive colour spaces
(`gray`, `rgb`, and `hsb`) zero values of the parameters
indicate that no light should be reflected from the
page (*i.e.*, the page is marked black), whereas for
subtractive colour spaces (`cmyk`) zero values indicate
that no light should be absorbed by the page (*i.e.*,
the page is left white). The double quote form
of colour specification supported by `dvips` is not
supported by `DVISep`.

A second form of `\special` command sup-
ported is `color push` followed by a colour specifica-
tion; this saves the current colour and sets the curent

---

[1] On systems with restricted filename length the spot
colour names are reduced using a heuristic rule which at-
tempts to create a recognisable name. `DVISep` does not
overwrite other files of the same name unless explicitly told
to, so other files which accidentally match the reduced spot
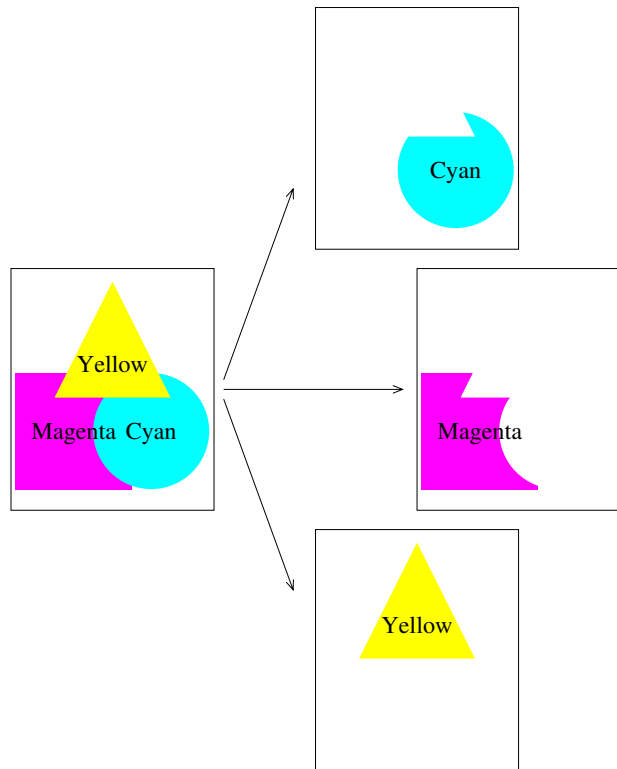colour name will be preserved.

Angus Duggan

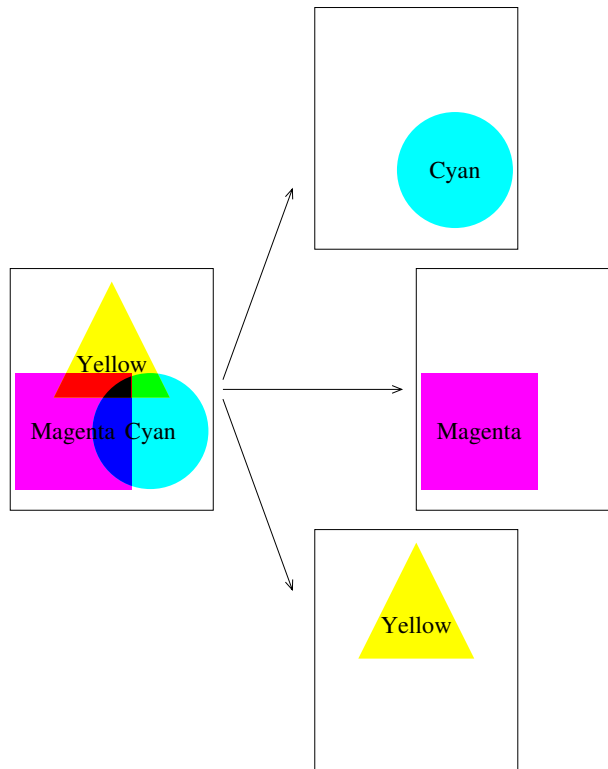**Figure 1**: Colour separation with knockout

Figure 2: Colour separation with overprinting

colour to the new colour. The command `color pop` sets the current colour to the colour last saved with a `color push` command

Colours specified by a colour space and parameters are converted to CMYK process colours before use. There is an option to DVISep which affects this conversion process. The `-u` flag turns on undercolour removal and black generation. RGB and HSB colours are initially converted to CMY colours. Undercolour removal removes an equal amount of each of the CMY components, and black generation adds that amount to the black component, yielding a CMYK colour.

Colours specified by names are looked up in a colour definition table; DVISep reads a default colour definition file when it initialises, which tells it the name, colour space, and parameters for named colours. Spot colours are also defined by the colour specification files. The format of these colour specification files is very simple—each line begins with a colour name, followed by either a colour space and numeric parameters, or the keyword `spot` and a single numeric parameter for spot colours. Spot colours may also have one of the optional keywords `overprint` or `knockout` at the end of the line. Comment lines in the colour specification file are indi-

cated by '`#`' as the first non-blank character on the line. Extra colour specification files can be loaded by passing the `-c` *file* option to DVISep. If a colour name in an extra colour specification file matches an existing name, the specification in the extra file is used. More than one extra colour specification files can be loaded by using several instances of the `-c` option.

The `-o` and `-k` options to DVISep indicate whether it is to overprint or knockout process colours; knockouts for process colours are the default. Whether spot colours overprint or knockout is determined by the colour specification file; they will normally knockout unless the colour specification contains the `overprint` keyword.

How DVISep works

Upon starting, DVISep reads a default colour specification file, and then processes its arguments, which may result in it reading more colour specification files. Colours specified in RGB or HSB space are transformed into CMY space when they are used, and then into CMYK through undercolour removal and black generation. The transformation from RGB and HSB to CMY is very simplistic; the algorithms in Foley, van Dam, Feiner and Hughes (Foley

et al., 1990) are used. If undercolour removal and black generation are turned on, the amount of black generated will be equal to the minimum value of the other colour components.

In general, colour transformation is more complicated than this; the representable range of colours on devices differs, depending on the printing process and inks used. The same colour values will produce noticeably different results not only on different devices, but also on different calibrations of the same device type and on different printing surfaces. This problem is beyond the scope of DVISep — a colour matching system is required to solve these problems.

DVISep makes several passes over the input DVI file. The first pass is used to build up a structure containing information about the pages in the file. The start and end of each page is noted, and lists of the spot colours and fonts used are generated. DVISep then makes one pass over the input file for each process colour and spot colour, to write out the separation files. On each output pass, the initial colour is set to black, and a colour stack is maintained by noting the `color push` and `color pop` special commands. As each page is scanned, DVISep looks for colour changing \specials and commands that mark the page (characters and rules). The action taken for page-marking commands depends on whether the current colour is a process colour or a spot colour, and whether the colour is overprinting or knocking out other colours. The name of the separation files is generated from the process and spot colour names. DVISep avoids writing out pages which do not contain any ink.

Process colour handling. When page-marking commands are found during process-colour separating, the current colour is examined to see if it is a process colour. If the current colour is a process colour with a non-zero component of the separation colour being generated, a colour support \special command will be issued which sets the colour to a shade of gray corresponding to the amount of the component present. For example, if the CMYK colour (0.87 0.68 0.32 0) is specified in the input file, a command to set the gray level to 0.13 will be issued for the cyan separation (remember that the gray parameters are inverted with respect to CMYK parameters), a command for a gray level of 0.32 will be issued for the magenta separation, and a command for a gray level of 0.68 will be issued for the yellow separation. The colour command will only be issued before the first page-marking command after each colour change.

If the current colour is a process colour with a zero component of the separation colour, or a spot colour, then DVISep needs to decide whether to knockout existing objects on the page. If the knockout flag is set for process colours, or the spot colour was specified with the `knockout` keyword (or no extra keyword at all), then a command is issued to set the current colour to white, and the page-marking commands are written to the separation file. Knockouts done in this way will only work if the imaging model of the output device is a paint-and-stencil model like PostScript,[2] where white areas painted over black areas erase them. If this imaging model is not assumed, knockouts have to be done by determining the difference of the shapes of the objects printed in the separation colour and the objects which are knocked out. The results of this process can not be represented in DVI format without generating custom fonts for the output resolution.

If the knockout flag for process colours is not set, or a spot colour is specified with the `overwrite` flag, then the page-marking commands up to the next change of colour are ignored. This allows different separations to have page-marking commands at the same position on the page, causing the inks to overprint and combine when printed. The page marking commands cannot be completely thrown away, however, because character and rule setting commands may move the current horizontal position. If characters are being ignored, DVISep reads and caches the TFM (TeX font metric) file for the current font, moving the current position right by the width of the character.

Spot colour handling. Spot colour separations are handled in a similar way to process colours, except that only one component is considered when deciding if the colour should be printed, knocked out, or ignored. The numeric parameter to the colour specification of a spot colour is a tint value, which indicates how much of the colourant should be applied, and hence the gray level for the following page-marking commands.

Background handling. The `background` colour command needs some special handling in DVISep. The last background command issued on each page is the one which takes effect (this is Rokicki's definition of the `background` special command), so this information must be stored in the page information during the initial scan. The background colour is treated much like other colours; if it is a process

---

[2] PostScript is a trademark of Adobe Systems Incorporated

colour, then appropriate background commands will be set for each separation file, depending upon the amount of the separation component in the background colour. If the background colour is a spot colour, then the background command will only appear in the spot colour separation file.

Using DVISep

DVISep normally takes a single input DVI file, and produces a DVI file for each process and spot colour used in a document. There are several other options which control DVISep's behaviour.

The -c, -k, -o, and -u flags have already been mentioned; the first flag names additional colour specification files to read, the next two flags indicate whether DVISep is to knockout or overprint process colours, and the last flag indicates whether to apply undercolour removal and black generation to RGB and HSB colours when converting to CMYK.

The -s *colour* and -p options are often used together; -s selects a single separation to output, and -p causes this separation to be written to standard output. This allows separations to be filtered through other programs from within scripts, without having to know the filename that DVISep would create for the separation colour.

Normally DVISep will not overwrite existing files; the -f option forces it to do so.

The usual DVIUtils options of -v for version information and -q for quiet running also apply.

Using DVISep with dvips. DVISep can be used with any device driver which supports Rokicki's colour \special commands. Input CMYK, RGB, and HSB colours (and named colours which are specified in these colour spaces) are converted to explicit gray scale commands.

It is important that drivers should know which separation is being output, so that halftone screens can be generated at the correct angles. There is no standard method of communicating this information to the driver; DVISep issues a new \special command to indicate the separation. This special command has the keyword separation followed by the separation name.

If the output of DVISep is being output through dvips, header files should be used to set up the separation screen angles. Process colour screens are traditionally at 15° for cyan, 75° for magenta, 0° for yellow, and 45° for black. The eye is very good at picking out vertical and horizontal features, which is why the least noticeable colour (yellow) has the angle nearest to orthogonal.

A header file that sets the screen angle to 15° without changing the screen frequency or spot function[3] might be defined as:

```
%!
%%DocumentProcessColors: Cyan
currentscreen exch pop 15 exch setscreen
```

The double quote form of colour specification is not supported by DVISep, because its parameter (an arbitrary PostScript string) does not give easily usable information about what colour is required.

DVISep supports an extended form of colour \special command, which allows spot colours to be specified from within TeX. This command takes the form spot *plate tint*, indicating the separation plate on which the colour is to appear, and its intensity. This colour command is not directly compatible with dvips; if the document is to be processed by dvips before separating (*e.g.*, for previewing) then spot colours should be specified by named colours, with a process colour approximation to the spot colour in the color.pro header file.

Limitations

DVISep has some limitations, which need to be borne in mind when using it. One of these limitations has been mentioned already; knockouts assume that printing in white can erase areas already printed in other colours. This limitation may be a problem for output drivers for a lot of devices, but fortunately not for PostScript drivers.

A more serious limitation from the user's point of view is that DVISep does not currently handle PostScript inclusions at all. There are reasons for this omission;

1. The \special commands used to insert PostScript code are output driver dependent.

2. The PostScript inclusion may reset the current colour. To a certain extent, this problem can be alleviated by putting the inclusion into each separation and redefining the colour-setting operators or transfer functions to set the colour to white for colours which should not appear on the current separation. This is not a total solution, because the inclusion can still access the original colour operators (and many applications produce PostScript which does this), and because overprinting cannot be done with inclusions.

---

[3] Some high-end PostScript RIPs can be configured to ignore user settings of frequency and spot functions, because the user's settings are not always appropriate to the final output device.

Both of these problems are insurmountable in the general case; there is no way of hiding the original colour operators completely, because the dictionary in which they reside is read-only, and itself is impossible to hide. The second problem is insurmountable because PostScript does not have the concept of transparency, except in the limited case of image masks. Sampled images may contain a mix of colours, and there is no facility for making some of the image pixels overwrite existing objects on the page, and making others leave the existing page untouched.

DVISep does not yet provide registration marks for aligning pre-separated plates, or taglines for identifying separations. I hope to add these soon.

DVISep does not attempt to trap the separations at all (trapping is used to reduce registration errors, by enlarging or reducing areas painted on different separations so that they have a very small overlap).

Conclusions

Colour separation is not necessary for many printers, especially desktop printers, which use their own colour rendering techniques to print continuous tone data. Separation is necessary when going to press with higher resolution work. Separating the DVI file is quite easy, but some assumptions have to be made about the imaging model which will be used. If film or plates are being produced, separating the DVI file may save time and effort, by removing the empty pages from the separation files before printing.

The second release of the DVIUtils programs (including DVISep) will be available for anonymous FTP.

References

Foley, James D, Andries van Dam, Steven K. Feiner, and John F Hughes. *Computer Graphics, Principles and Practice*. Addison-Wesley, Reading, MA, USA, second edition, 1990.