Computer Science Department
University College
Australian Defence Force Academy
Campbell, ACT 2600
Australia

Computer Science Department Report CS24/91

Aurora: Colour Separation with PostScript Devices

T. Graham Freeman

October 1991
Revised July 1994

Summary: For material to be offset printed, the masters need to be prepared as colour separations, one for each coloured ink. Production of the masters is a time consuming task, often performed by hand. A method is proposed to enable separations to be automatically prepared if the source material is available as a colour PostScript file. The method uses any black-and-white PostScript printer, and two header files, the first defining procedures to be used in preparing the separation, and the second selecting the colour of the separation desired. The method works with any colour Level 1 PostScript documents.

Background

Offset printing in colour is most frequently achieved by separately printing the chosen inks onto the final piece of paper. Separate plates are made up for each of the colours, and they must be carefully aligned or *registered* to reproduce the desired combination of colours. If you have an original illustration or manuscript in colour, the printer must manually separate the colours into the separate plates for reproduction. This is a slow and expensive process. Printers prefer to be given the originals already separated into the component colours.

If you are producing the material on a computer for later reproduction by offset printing, and colour is needed, there is currently little support for automatically producing the colour separations. The most common output devices are PostScript printers. Some of these will provide colour output, but although these are good for seeing the final colours, they are generally unsuitable for use by an offset printer. A system is needed for taking the computer-stored document and producing the colour separations automatically. Some specialized hardware is available to do this, but it can be expensive.

Colour printing is performed to meet two end requirements: firstly, one or more carefully chosen colours are used along with black to give a greater impact to the final document; secondly, a standard set of colours is used to give full colour reproduction.

In the first case, the highlight colours are frequently used for headings or to draw attention to features of the content. They are often physically separate from the black text on the page, to the extent that a printer can take original manuscript, and by masking out the appropriate areas, select those parts of the document which are to be produced in each colour. For this to be effective, a strong contrast is needed in the original master for each of the highlight colours, so it is usual for the manuscript to be provided in all-black, with separate instructions on which colour to use for which parts.

In the second case, the standard colours used in printing are cyan (a blue similar to sky colour), yellow, and magenta (a strong pink-red). Because the combination of the three inks usually comes out as a dark brown or blue, black is used as a fourth colour, being superimposed with the other inks when dark shades are required, and used alone for full black. Full colour reproduction is necessary in cases where photographs or images are wanted, or when the range of colours to be reproduced is greater than the small range of colours of the first case.

PostScript Approaches to Colour Separation

A system for producing colour separations from PostScript documents has been described by Adobe (1990). New features were added to the PostScript language to enable special-purpose hardware to generate separations automatically. Alternatively, Adobe also advocate use of a comment convention within PostScript documents whereby each change of colour is preceded by an appropriate comment. These comments can then be recognized by a program running in the host, which will filter out those parts of the document that are of a different colour from the current separation being produced, and spool only those parts of the selected colour.

This strategy is clearly directed at the first case above, where the colour usage within a document is distinct. It also relies on the document having been produced with each colour usage encapsulated, so that colour change is easily recognized by spooling software which does not need to examine the actual PostScript instructions.

While this system is very simple, it is clear that these restrictions severely limit the applicability of this scheme. The greatest problem is the assumption of independence of the colours. Colour images cannot be produced under this scheme. Thus there is a need for a system that has more general application with fewer of these assumptions about the nature of the document.

In an earlier paper, Adobe (1989) advocated the use of special procedures and redefined operators to filter out the colour of the desired separation. This is intended to enable full colour as well as highlighting. The approach restricts the PostScript that can be supplied, in that operators requesting current colour state of the printer cannot be used, and only one of the several forms of colour image can be processed. However, it is not based on embedded comments within the document which are ignored by the PostScript printer, but rather uses the properties of the PostScript language itself.

PostScript is a programming language for describing the page layout of a document. The document description is interpreted within the printer by a computer, which builds up a bitmap representing the final page output, and outputs it to the paper before commencing the next page. Because the language is interpreted, it is possible to redefine the fundamental operators of the language within a document itself.

The approach that we will adopt is similar, being based on redefining the colour operators to enable output on a black-and-white printer. It is intended to be sufficiently general to accept any PostScript file, circumventing the restrictions imposed by Adobe (1989). The package of routines described here is named Aurora, after the coloured phenomenon in the sky near the poles, or the colour of the sky at dawn.

Aurora Design

Many colour PostScript printers produce colour by laying down each of the component colours in succession by passing the paper through a number of times. In theory, it should be possible to pass separate pieces of paper through to generate the separations without any extra processing. Such an approach would not be satisfactory, because the offset printer needs high contrast originals to make the plates, and yellow-on-white or cyan-on-white originals are very difficult to work with. The intention with Aurora is that it will produce black-and-white separations for each of the intended colours.

Because the output will be in black-and-white for each separation, it is not necessary to have a colour PostScript printer to produce the separations; a black-and-white printer is all that is required. Such a piece of hardware does not need to be capable of interpreting any of the colour PostScript operators, because all will be redefined in terms of existing black-and-white operators.

To use Aurora, first a prelude must be sent to the printer, redefining the operators that influence the colour of output. Then a second header is sent which selects the colour of the required separation. Finally, the document itself is sent.

The PostScript definition has undergone a major extension, known as Level 2, in which new methods of describing colour are introduced. When the design of this software was undertaken, almost all printers and software for colour production on the market used the operators of Level 1 and its extensions. For this reason, colour separation has been confined to output that does not use the most recent Level 2 features.

The operators that need to be redefined for Level 1 and its extensions are:

1. currentgray
2. setgray
3. currentcmykcolor
4. setcmykcolor
5. currentrgbcolor
6. setrgbcolor
7. currenthsbcolor
8. sethsbcolor
9. initgraphics
10. setcolorscreen

11. currentcolorscreen
12. setcolortransfer
13. settransfer
14. currentcolortransfer
15. setblackgeneration
16. currentblackgeneration
17. setundercolorremoval
18. currentundercolorremoval
19. colorimage
20. image
21. gsave
22. grestore
23. grestoreall
24. save
25. restore

The first nine operators are fairly straightforward. The colour separation software needs to keep track of the current colour, and set the actual output colour to white unless the requested colour should have a component of the current separation colour. Every change of colour has to be recorded in the local colour state of Aurora, so that later requests for current colour can be honoured in a transparent way. The colour state is held as values of cyan, magenta, yellow and black (CMYK) because these are the common output colours for hardcopy. Routines are needed to convert the other colour specifications to this colour space:

rgb2cmyk	Red, Green, Blue (RGB) to CMYK
hsb2rgb	Hue, Saturation, Brightness to RGB

It is not possible to convert in the reverse direction, from CMYK to RGB or HSB, since the method for adding black for dark values is a one-way function provided via "setblackgeneration" and "setundercolorremoval", which has no explicit inverse. An approximate conversion scheme can be implemented that ignores black generation and under-colour removal, to handle the unlikely situation where a document contains a "setcmykcolor" followed by a "currentrgbcolor" or "currenthsbcolor". It should be noted that at least one hardware implementation of PostScript gives very strange RGB values if this is attempted! In Aurora, it is safest to store both RGB and CMYK versions of the current colour, in order to respond to current colour requests in either system most accurately. The conversion scheme from CMYK to RGB suggested by Adobe (1990) is used to keep the local record of RGB approximately correct. A function is needed:

rgb2hsb	RBG to Hue, Saturation, Brightness

to enable "currenthsbcolor" requests to be satisfied.

Operators ten to fourteen in the above list are simple colour extensions of their black-and-white equivalents, allowing separate control of transfer function and screen for each of the four printing primary colours. A local record must be kept of the values or procedures set in case they are requested in a following PostScript program. Only the screen or transfer function for the current separation colour will be passed through to the PostScript hardware.

Operators fifteen to eighteen cover the addition of black to the other primaries when a dark colour is requested, and the removal of part of the other colours to compensate for the addition of the black. Each of these takes a single procedure which must be held locally and not passed through to the PostScript printer. These procedures may be requested from within

a following PostScript program. If they are specified in an output document, the procedures must be applied in all conversions from RGB to CMYK. By default, Aurora will assume that there is no under-colour removal, and that black generation takes place when the lowest of the cyan, magenta, and yellow values is less than 0.25, varying linearly between zero and one as the lowest of these other primaries ranges from 0.25 down to 0.

Colour Images

The PostScript operator that requires most effort to implement is "colorimage". While having many operands in common with the black-and-white "image" operator, there are several additional pieces of information for colour output. Each of the possibilities must be carefully considered in order to filter out the information relevant to the current colour separation.

The first operand indicates how many colours are provided in the image, with one implying a black-and-white image, three implying RGB colours, and four implying CMYK. The second operand indicates whether a single procedure will generate a string containing the colour values, or a separate procedure exists for each colour, generating a string for the respective colour. The "image" or the "colorimage" operators repeatedly execute the procedure(s) until the required number of values for the given image size have been generated. Each combination of these operand possibilities must be handled separately in Aurora, using the "image" operator instead of "colorimage", with the operand stack suitably modified.

In processing each case, it is necessary to execute the supplied procedure(s) in the way that the "colorimage" operator would, because sometimes, data follows on the input stream that is read by the procedures, or data already exists on the stack. The behaviour of the supplied PostScript program must not be compromised in changing the colour of the output. Executing the procedure(s) will generate one or more strings on the stack. The "image" operator that we substitute for "colorimage" will either use one of these strings, or use a shorter string made up of information regularly scattered throughout a longer string. It is necessary within "colorimage" processing to shuffle the string(s) generated before "image" sees the single resulting string that is intended for it. Thus, "colorimage" must build a new procedure from the one(s) supplied, performing the required manipulations.

In the cases where a single longer string is generated for "colorimage", it is not wise to simply shuffle the desired values down to one end of the string for use by "image". When a string is placed on the stack, only a pointer to the string is put there; a single copy of the string is held in virtual memory, so altering the string will cause every pointer to it, whether deeper on the stack or in dictionaries, to be affected. If the "colorimage" is to be executed more than once, corruption of the image could occur if the string were altered within the "colorimage" procedure. It is therefore necessary to build a new string with the selected information from the single longer string generated by "colorimage".

The case of one colour can be readily translated into use of "image" for output. However, care is needed if the current separation is not black. The image must still be output, because there may be graphics beneath that the image will obscure. Thus, the image must be put out with all white values, and a string must be constructed of the same length as that generated by the "colorimage" procedure operand, filled with white values.

RGB colour images require extra processing, because each colour value must be obtained and converted to CMYK before the colour component in the current separation can be determined. This value is then placed in the new string for processing by "image".

Another complication is the number of bits per value. If this is a number other than eight, appropriate masking and shifting must be performed to get values from the string(s) on the stack to be converted to CMYK or to store scattered values contiguously in another string.

Grayscale Images

Because the "image" operator produces black-and-white output regardless of the currently selected colour, it also requires special processing, in the same way that single colour output from "colorimage" was handled above. Once again, the image must be processed whether the current separation is black or some other colour. If non-black, the image output must be all-white, to obscure any existing output that the image should superimpose.

The "imagemask" operator does not need to be considered, because it produces output in the current colour.

Implementation Considerations

From the above description, it can be seen that several items must be held locally by Aurora, in order to respond to various colour state requests, and in order to ensure that colour changes have the correct effect on the current separation output. This information needs to be held in such a way that the names chosen for them will not conflict with any usage in whatever PostScript that may follow. A dictionary is created by Aurora to hold this local information as well as local procedures for colour conversion, bit shuffling, etc. The only name defined by Aurora that could potentially conflict with following PostScript instructions is the name of Aurora’s dictionary itself. Procedures introduced to redefine the operators above must use this dictionary to access the information needed, but must also be careful that the dictionary is not left on the dictionary stack whenever user-supplied procedures are being executed.

Implementation of "colorimage" requires that a new string be built up containing the information that "image" will use for the separation output. If this string is created every time "colorimage" is executed, virtual memory will be wasted, since in Level 1 Postscript such usage cannot be recovered. To minimize risk, a single string is kept in the Aurora dictionary for "colorimage" output. If it is not long enough, it is replaced with one of sufficient length. If it is too long, the substring of correct length is extracted with the "getinterval" operator.

Because "grestore" or "grestoreall" can alter the current colour, it is necessary to keep a local stack of internal colour usage, so that after a "grestore", requests for current colour or gray-scale can be correctly responded to. This stack is implemented as an array containing the essential local information: CMYK colour, RGB colour, halftone screen, black generation, under-colour removal, and transfer function. The array has to be set aside with some fixed size, so "gsave" operations need to check for overflow, and if necessary create a larger array, copying the saved elements across. In this way, the number of "gsave" operations should not

be limited by Aurora.

Saving local information in the local stack must be done allowing for the fact that PostScript does not actually store the information of a composite variable like an array in more than one place; only a pointer is placed with the "def" operator. Thus it is necessary to actually copy the contents of arrays into the parts of the array element representing the top of the stack, and to copy in the reverse direction upon "grestore" in those cases where the information is held in arrays.

The need for local versions of "save" and "restore" is less obvious. Because "grestoreall" will restore the first graphics state stored, or the most recent "save" state, it is necessary at each "save" to record the state in such a way as to distinguish it from that saved by a "gsave" operation. Then "grestoreall" will peel away graphic states from the local stack until either the stack is exhausted, or a save state from a "save" is encountered.

The numbers representing colour in PostScript gray-scale and RGB values have the opposite sense to their usage in CMYK. This must be allowed for in the "colorimage" implementation when converting to black-and-white for final output.

Custom Colours

The scheme described so far enables almost any PostScript document description to be run on a black-and-white printer to generate the separations for cyan, magenta, yellow and black. In practice, to reduce the number of inks and hence the cost, it is often desirable to choose other coloured inks. Features have been added to Aurora to facilitate this requirement, but with greater restrictions on the type of PostScript operators that can be used, and which colours can be chosen.

An ink colour is defined in terms of cyan, magenta, yellow and black values, and will be referred to here as custom colours (see Appendix for mappings). Any colour to appear in the output can be considered a point in the four dimensional CMYK space, because there are four independent components to each colour. In practice, a three dimensional space is more commonly used, because high values of black make values of other components undetectable.

When using Aurora, a custom colour is specified at the start. This colour specifies a point in CMY space of the separation being produced. When the current colour in the document lies on the line from the origin to the point for a selected custom colour, then that colour is printed by Aurora. The amount of that colour is determined by where along this line the colour lies. In this way, only document colours matching the custom colour will be output.

This method of selecting colours for output when using custom colours requires that the numbers specifying the custom colour almost precisely match the colour used within the document. If the colour does not match, it will produce no output, in contrast with the normal process colours, where the the component of the current process colour determines the amount of output to produce, and an imprecise match in colours will still give a strong colour.

In principle, the use of special inks could be used to achieve some form of colour output with images. Of course full colour would not be possible, but for some applications, enough

colour variety could be present to enliven an otherwise gray-scale image. This extension has not been made in Aurora at this stage; all graphics output apart from colour image is catered for.

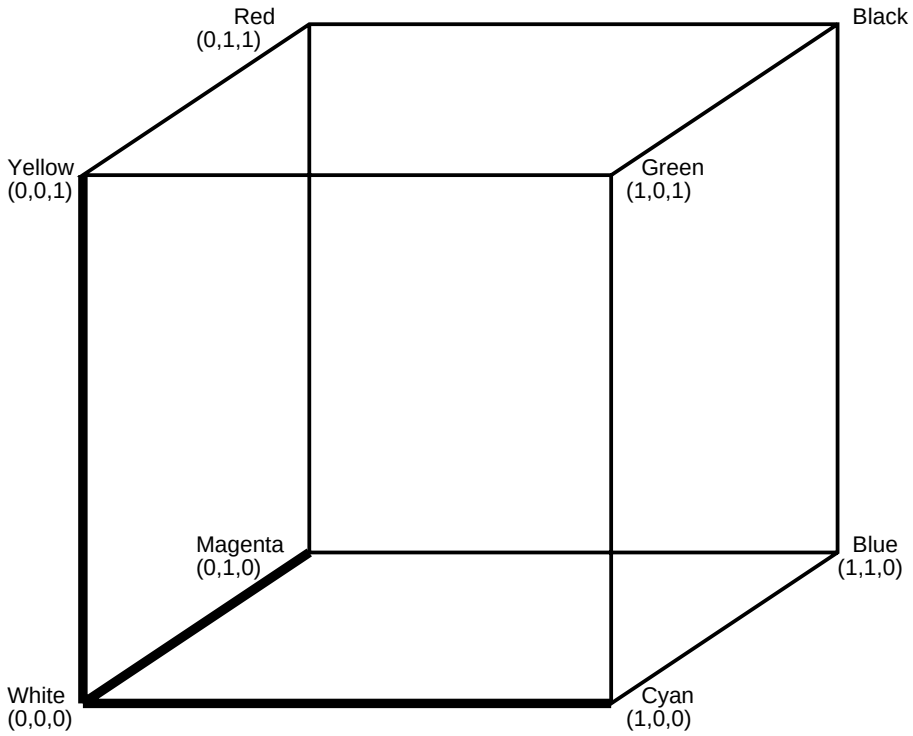


Figure 1: Colour space of cyan, magenta, yellow primaries. A colour can be considered to be a point in this cube.

Restrictions

PostScript is a general purpose language designed for output of text and graphics. Our aim has been to provide colour separations regardless of what legal PostScript program is used. With the changing specification of PostScript, it is difficult to achieve full support for all colour features of the language. Most of the restrictions that apply are unlikely to be a problem for practical output from document production packages. The worst of the restrictions apply to use of custom colours.

Colour images are restricted to having 1, 2, 4 or 8-bit values, consistent with the Level 1 PostScript colour extensions, to eliminate difficulties in having values spanning byte boundaries. Level 2 PostScript allows 12-bit values; Aurora could be extended to handle this case, but with other Level 2 features not supported and slow reliance of commercial software on Level 2, this is not seen as necessary in the immediate future.

The major Level 2 features that are not allowed for are:
CIE colour specification operators
"currentcolorrendering"
"setcolorrendering"
colour space and associated operators
"currentcolor"

"setcolor"
"currentcolorspace"
"setcolorspace"
"currentoverprint"
"setoverprint"
"setpattern"

operator extensions that use a dictionary instead of the conventionalset of operands

"image"
"currenthalftone"
"sethalftone"

graphics state extensions

"setgstate"
"currentgstate"
"gstate"

If the "setcolortransfer" or "setcolorscreen" operators are used, colour separations with custom coloured inks are not possible, because there is no way to determine which transfer function or screen to use.

As has been mentioned above, colour images must be based on the normal colour primaries and cannot be based on custom colours.

Conclusion and Examples

The scheme outlined in this paper allows general colour PostScript to be despatched to a black-and-white printer with two header files, the first defining the procedures needed, and the second selecting the separation to be output. This must be done independently for each of the desired separations. There is no restriction in the content of the PostScript, apart from the lack of support for Level 2 features.

The resulting separations will be true to the PostScript output model, in that if the output sequence places one colour over the top of another, only the most recent colour will appear in the final output. Thus, if special effects are desired with colours being mixed in different areas by overlaying, this will not be what is observed. The areas of mixing will have to be specified with their own colour, as PostScript requires.

An example is shown in Figure 2 of simple coloured graphics and in Figure 3 of an RGB colour image separated into the four usual components. The colour image used a single procedure, generating consecutive red, green and blue values.

Aurora can be quite slow, particularly for colour image processing. Substitution of internally executed operators by interpreted procedures can only have a detrimental effect on performance, particularly on older machines. Figure 3 has been timed taking 101 seconds to view on an X terminal with the PostScript previewer *ghostscript*, and takes 31 minutes to print on an Apple LaserWriter Pro 810. However, on a very old Apple LaserWriter, the same page takes 103 minutes! This represents an extreme case. It is unusual to print multiple images on the one page, and images are the slowest items to output with Aurora. (The images in Figure 3 are each 200x200 pixels.)

Simple graphics are handled much more efficiently. In Figure 2, three overlapping circles are drawn, of colour brown, yellow and green, each with a thin black outline, and a black square superimposed. Although the colours do not reproduce satisfactorily in black-and-white printed versions of this paper, the separations can be seen to reflect the final colours according to the PostScript model, rather than mixing the colours from the earlier circles. This illustration takes two seconds to draw on an X terminal using *ghostscript*, 7.4 seconds on an Apple LaserWriter Pro 810, and 33 seconds on an old LaserWriter.

In this example, the shape is drawn with the instructions:

```
   %       x y  Circ  -
   %  Draws a circle at (x,y) of radius 30.
/Circ{ newpath 1 index 1 index moveto 30 0 rmoveto  30 0 360 arc
   gsave fill grestore  0 setgray stroke}def
/Shape{
   .7 .5 0 setrgbcolor  35 50 Circ    %brown
   0 0 1 0 setcmykcolor 55 50 Circ    %yellow
   0 1 0 setrgbcolor    45 30 Circ    %green
   0 setgray 15 10 moveto
   60 0 rlineto 0 60 rlineto -60 0 rlineto closepath stroke
   1 1 1 1 setcmykcolor 0 -5 moveto
   90 0 rlineto 0 90 rlineto -90 0 rlineto closepath stroke
} def
```

We are using *setrgbcolor* to set the brown and the green, but *setcmykcolor* to set the yellow. A square is superimposed on the circles in black (0 setgray). In order to assist in aligning the separations in the offset printing process, a square outline has been drawn on all separations. This is achieved by using a colour 1,1,1,1 CMYK.

We could print such an illustration either by separating into the CMYK primaries, or by using custom colours to match each of the colours appearing in the drawing. If we wanted separates based on the usual CMYK process colours [Figure 2 (b) to (e)], the Cyan separation would be obtained by following Aurora with the line:

```
_c+stat begin /this 0 def end
```

The Magenta separation requires the line

```
_c+stat begin /this 1 def end
```

and the yellow separation the line

```
_c+stat begin /this 2 def end
```

Although it is not sensible to mix custom colours and process colours with Aurora, Figure 2(f) shows what would result if we intended to use a brown custom colour. To select the brown colour, Aurora must be followed by a line:

```
_c+stat begin /this [.7 .5 0 rgb2cmyk ] def end
```

or alternatively

```
_c+stat begin /this [.3 .5 1 0] def end
```

In general any custom colour can be specified by placing its CMYK values between the [] brackets.

If we wanted to use brown, green and yellow colours in the printing process rather than cyan, magenta and yellow, then they would need to be selected individually in separate runs of Aurora with the colours precisely specified as custom colours in [] brackets, rather than as process colours, eg

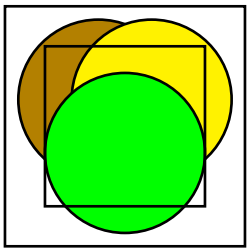
```
_c+stat begin /this [0 0 1 0] def end
```

for a yellow custom colour, and

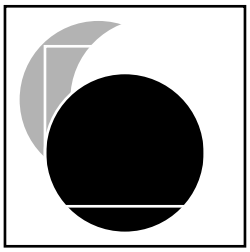
```
_c+stat begin /this [1 0 1 0] def end
```

for the green. These would give the results shown in Figures 2(g) and (h).

When colours are to be separated with Aurora, best results are likely to be achieved if different colour screens are used for each primary colour. The PostScript operator *setcolorscreen* can be used to set different dot positions for halftones of the different colours. If this operator is not used, the same screen will be used for each separate, which will mean that dots for the separates will be superimposed, reducing colour reproduction quality.

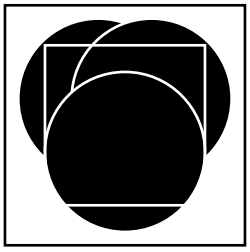


(a) Full colours

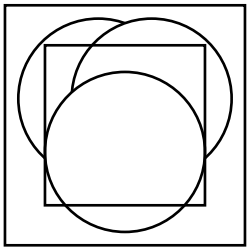


(b) Cyan separation

(c) Magenta separation

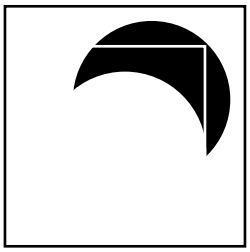


(d) Yellow separation

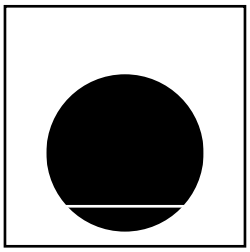


(e) Black separation

(f) Brown custom colour

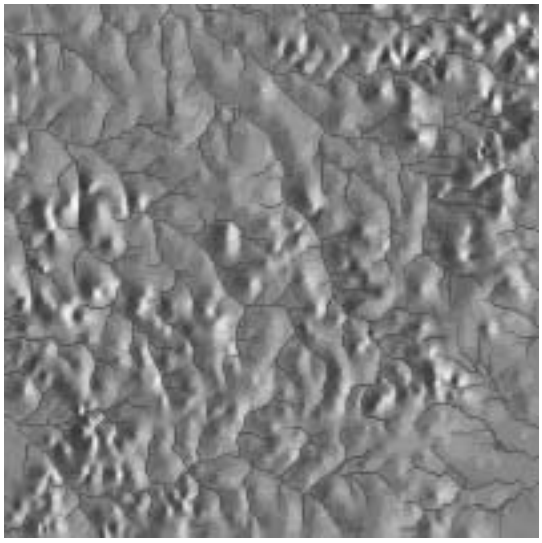


(g) 0,0,1,0 custom colour

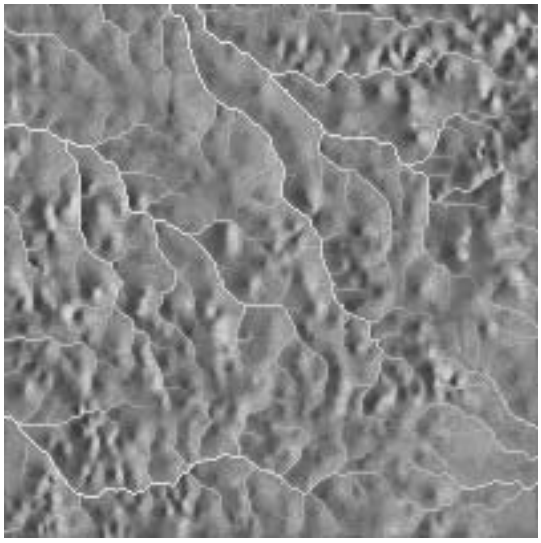


(h) 1,0,1,0 custom colour

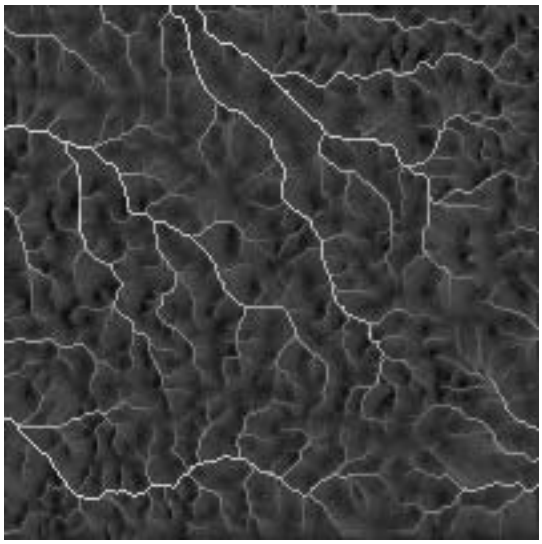
Figure 2: Separations of a simple multi-colour drawing. (a) the original drawing in brown, yellow, green and black, (b), (c), (d) and (e) CMYK separations, (f) brown custom colour, (g) and (h) yellow and green custom colours.



Cyan separation



Magenta separation



Yellow separation

Black separation

Figure 3: CMYK separations of a synthetic RGB terrain image showing shaded relief (black through to yellow-orange) and drainage network (green through to blue). Although some of the colours are dark, only a few in the upper right corner are dark enough to cause black generation to be invoked, as seen in the nearly empty black separation.

References

Adobe Systems Inc (1989) Proposal for Color Separation Conventions for PostScript Language Programs. Technical Note 5044, PostScript Developer Support Group, Adobe Systems.

Adobe Systems Inc (1990) PostScript Language Reference Reference Manual, Second Edition. Addison-Wesley.

Appendix — Pantone Values

In the printing industry, colours are very commonly specified as PANTONE® inks. The following table gives a guide to the CMYK values (percentages) corresponding roughly with various PANTONE colours.

Pantone	C	M	Y	K	Pantone	C	M	Y	K	Pantone	C	M	Y	K	Pantone	C	M	Y	K
100	0	0	50	0	157	0	44	72	0	214	0	100	37	15	271	44	39	0	0
101	0	0	74	0	158	0	59	84	0	215	0	84	39	34	272	55	48	0	0
102	0	0	100	0	159	0	63	100	15	216	0	75	39	48	273	78	100	0	11
103	0	0	100	25	160	0	59	100	39	217	0	37	0	0	274	78	100	0	34
104	0	0	100	37	161	0	50	100	59	218	0	63	0	0	275	78	100	0	41
105	0	0	100	50	162	0	18	25	0	219	0	78	0	0	276	74	81	0	50
106	0	0	69	0	163	0	37	48	0	220	0	100	25	25	277	34	11	0	0
107	0	0	74	0	164	0	48	69	0	221	0	100	25	39	278	41	18	0	0
108	0	0	100	0	165	0	59	84	0	222	0	100	21	55	279	66	39	0	0
109	0	15	84	0	166	0	63	100	0	223	0	50	0	0	280	100	72	0	25
110	0	18	84	11	167	0	59	100	25	224	0	66	0	0	281	100	72	0	41
111	0	18	100	34	168	0	55	100	55	225	0	75	0	0	282	100	69	0	55
112	0	18	100	41	169	0	25	25	0	226	0	100	0	0	283	39	0	0	0
113	0	11	66	0	170	0	41	48	0	227	0	100	0	25	284	48	11	0	0
114	0	11	69	0	171	0	55	66	0	228	0	84	0	44	285	78	41	0	0
115	0	11	74	0	172	0	63	75	0	229	0	78	21	59	286	100	59	0	11
116	0	21	84	0	173	0	69	100	11	230	0	39	0	0	287	100	66	0	18
117	0	25	100	18	174	0	66	100	41	231	0	63	0	0	288	100	63	0	37
118	0	25	100	34	175	0	59	78	55	232	0	74	0	0	289	100	59	0	55
119	0	18	100	50	176	0	29	21	0	233	15	100	0	0	290	34	0	0	0
120	0	11	55	0	177	0	48	44	0	234	0	100	0	37	291	44	0	0	0
121	0	15	66	0	178	0	63	55	0	235	0	100	0	44	292	59	18	0	0
122	0	25	75	0	179	0	74	81	0	236	0	41	0	0	293	100	55	0	0
123	0	37	84	0	180	0	72	75	18	237	0	63	0	0	294	100	50	0	21
124	0	34	100	11	181	0	69	75	50	238	11	72	0	0	295	100	50	0	39
125	0	29	100	34	182	0	34	15	0	239	21	100	0	0	296	100	44	0	66
126	0	29	100	39	183	0	50	29	0	240	25	100	0	0	297	44	0	11	0
127	0	11	50	0	184	0	69	44	0	241	25	100	0	15	298	63	0	0	0
128	0	15	63	0	185	0	81	72	0	242	0	84	0	55	299	72	0	0	0
129	0	21	72	0	186	0	81	72	11	243	0	37	0	0	300	100	37	0	0
130	0	34	100	0	187	0	81	69	29	244	11	44	0	0	301	100	37	0	25
131	0	34	100	15	188	0	74	63	50	245	18	59	0	0	302	100	25	0	50
132	0	29	100	37	189	0	41	18	0	246	29	100	0	0	303	100	0	0	69
133	0	25	100	55	190	0	55	29	0	247	34	100	0	0	304	29	0	11	0
134	0	15	44	0	191	0	72	41	0	248	39	100	0	21	305	41	0	11	0
135	0	25	59	0	192	0	84	63	0	249	29	78	0	48	306	63	0	11	0
136	0	34	74	0	193	0	81	63	18	250	0	25	0	0	307	100	0	0	29
137	0	39	81	0	194	0	81	55	39	251	21	44	0	0	308	100	0	15	50
138	0	41	84	0	195	0	72	55	55	252	29	55	0	0	309	100	0	29	55
139	0	41	100	29	196	0	34	11	0	253	44	84	0	0	310	41	0	18	0
140	0	34	100	55	197	0	48	18	0	254	50	100	0	0	311	50	0	25	0
141	0	18	48	0	198	0	74	39	0	255	44	81	0	39	312	78	0	25	0
142	0	29	72	0	199	0	100	63	0	256	15	29	0	0	313	100	0	18	15
143	0	37	75	0	200	0	100	63	21	257	21	41	0	0	314	100	0	21	29
144	0	48	100	0	201	0	100	63	39	258	44	72	0	0	315	84	0	29	48
145	0	48	100	15	202	0	100	63	48	259	50	100	0	21	316	100	0	39	59
146	0	41	100	39	203	0	39	11	0	260	50	100	0	39	317	21	0	18	0
147	0	34	100	66	204	0	59	11	0	261	48	100	0	48	318	39	0	29	0
148	0	18	41	0	205	0	72	11	0	262	37	84	0	63	319	48	0	34	0
149	0	25	50	0	206	0	100	44	0	263	15	21	0	0	320	81	0	41	0
150	0	37	66	0	207	0	100	44	25	264	29	37	0	0	321	84	0	44	21
151	0	48	78	0	208	0	81	39	41	265	50	59	0	0	322	100	0	48	29
152	0	55	100	0	209	0	75	39	50	266	72	100	0	0	323	84	0	48	41
153	0	48	84	25	210	0	44	11	0	267	75	100	0	0	324	25	0	21	0
154	0	48	84	39	211	0	55	11	0	268	75	100	0	29	325	50	0	37	0
155	0	15	34	0	212	0	66	15	0	269	74	100	0	48	326	69	0	48	0
156	0	25	44	0	213	0	100	37	0	270	34	29	0	0	327	84	0	55	21

328	84	0	55	29	395	15	0	75	0	462	63	63	100	0	529	34	44	0	0
329	84	0	55	41	396	18	0	84	0	463	59	63	100	0	530	25	37	0	0
330	81	0	55	50	397	18	0	100	18	464	48	63	100	0	531	18	25	0	0
331	21	0	21	0	398	11	0	100	29	465	25	37	55	0	532	84	81	72	11
332	25	0	25	0	399	0	0	100	44	466	18	29	44	0	533	84	78	48	0
333	39	0	37	0	400	0	11	11	21	467	15	21	39	0	534	78	69	34	0
334	78	0	66	0	401	0	11	21	29	468	11	15	29	0	535	44	34	11	0
335	78	0	69	15	402	0	11	21	39	469	0	50	84	59	536	37	25	11	0
336	78	0	72	34	403	0	11	25	48	470	0	55	84	39	537	29	21	11	0
337	29	0	29	0	404	0	15	29	55	471	0	55	100	25	538	21	15	11	0
338	44	0	39	0	405	0	18	41	72	472	0	39	50	0	539	100	50	0	66
339	66	0	66	0	406	0	11	15	25	473	0	29	41	0	540	100	48	0	48
340	74	0	69	0	407	0	11	15	34	474	0	21	29	0	541	100	50	0	37
341	81	0	72	21	408	0	18	18	39	475	0	18	25	0	542	66	25	0	15
342	84	0	74	41	409	0	21	25	48	476	63	72	100	0	543	50	15	0	11
343	81	0	75	48	410	0	25	29	55	477	59	78	100	0	544	39	0	0	0
344	29	0	34	0	411	0	34	44	72	478	55	78	100	0	545	34	0	0	0
345	39	0	41	0	412	0	37	69	81	479	34	44	48	0	546	81	21	0	75
346	50	0	50	0	413	0	0	11	25	480	21	34	37	0	547	84	29	0	72
347	74	0	81	0	414	0	0	15	37	481	15	25	25	0	548	100	25	0	63
348	81	0	100	15	415	0	0	18	44	482	11	18	18	0	549	55	0	0	37
349	78	0	100	37	416	0	0	21	50	483	0	72	72	59	550	41	0	0	29
350	66	0	84	59	417	0	0	29	63	484	0	78	75	37	551	34	0	0	21
351	18	0	25	0	418	0	0	37	72	485	0	100	81	0	552	21	0	0	15
352	25	0	34	0	419	39	0	39	81	486	0	48	44	0	553	55	0	55	69
353	37	0	41	0	420	0	0	0	21	487	0	39	34	0	554	72	0	66	59
354	59	0	81	0	421	0	0	0	34	488	0	34	25	0	555	69	0	66	55
355	66	0	84	0	422	0	0	0	39	489	0	21	18	0	556	44	0	37	34
356	69	0	100	21	423	0	0	0	48	490	0	63	63	66	557	34	0	29	21
357	63	0	100	48	424	0	0	0	59	491	0	69	59	50	558	25	0	21	15
358	29	0	44	0	425	0	0	0	72	492	0	72	63	41	559	18	0	15	11
359	39	0	55	0	426	0	0	0	81	493	0	50	29	11	560	59	0	59	63
360	48	0	75	0	427	0	0	0	18	494	0	39	18	0	561	63	0	55	48
361	55	0	81	0	428	0	0	0	29	495	0	29	15	0	562	66	0	55	37
362	59	0	100	15	429	11	0	0	39	496	0	21	11	0	563	44	0	34	11
363	59	0	100	25	430	11	0	0	48	497	0	50	63	72	564	37	0	29	0
364	59	0	100	41	431	18	0	0	63	498	0	50	55	55	565	25	0	21	0
365	15	0	37	0	432	29	0	0	74	499	0	55	59	48	566	18	0	15	0
366	21	0	48	0	433	37	0	0	81	500	0	41	25	18	567	59	0	59	59
367	29	0	59	0	434	11	15	15	0	501	0	34	18	0	568	72	0	63	41
368	50	0	100	0	435	18	21	21	0	502	0	25	15	0	569	78	0	63	25
369	55	0	100	0	436	29	29	34	0	503	0	18	11	0	570	44	0	37	0
370	50	0	100	29	437	41	41	44	0	504	66	84	100	15	571	34	0	29	0
371	41	0	100	50	438	66	63	84	11	505	63	81	84	0	572	25	0	21	0
372	15	0	39	0	439	74	72	100	11	506	59	81	78	0	573	18	0	15	0
373	21	0	50	0	440	75	72	100	11	507	21	50	29	0	574	39	0	72	59
374	25	0	59	0	441	11	0	11	18	508	11	41	18	0	575	50	0	81	50
375	39	0	74	0	442	11	0	11	29	509	0	37	15	0	576	50	0	81	41
376	48	0	100	0	443	15	0	15	39	510	0	29	11	0	577	29	0	50	18
377	44	0	100	29	444	15	0	15	48	511	66	100	66	0	578	25	0	44	11
378	39	0	100	55	445	21	0	21	63	512	50	81	34	0	579	25	0	39	0
379	15	0	59	0	446	18	0	25	72	513	44	78	0	0	580	18	0	29	0
380	18	0	69	0	447	21	0	37	75	514	18	50	0	0	581	0	0	84	66
381	25	0	81	0	448	66	59	100	0	515	11	41	0	0	582	18	0	100	48
382	34	0	84	0	449	63	59	100	0	516	0	34	0	0	583	34	0	100	21
383	25	0	100	25	450	63	55	100	0	517	0	25	0	0	584	18	0	74	11
384	21	0	100	37	451	37	29	50	0	518	66	84	66	0	585	18	0	63	0
385	0	0	78	55	452	25	21	41	0	519	63	84	41	0	586	15	0	50	0
386	11	0	55	0	453	21	15	34	0	520	59	84	25	0	587	11	0	44	0
387	15	0	72	0	454	15	11	25	0	521	34	50	0	0	1345	0	18	48	0
388	18	0	74	0	455	0	25	100	63	522	25	41	0	0	1355	0	25	55	0
389	25	0	75	0	456	11	25	100	44	523	15	29	0	0	1365	0	37	72	0
390	25	0	100	11	457	0	25	100	37	524	11	21	0	0	1375	0	41	72	0
391	18	0	100	34	458	18	18	69	0	525	74	100	55	0	1385	0	48	100	15
392	11	0	100	48	459	11	11	50	0	526	66	100	18	0	1395	0	44	100	41
393	11	0	50	0	460	11	11	41	0	527	66	100	0	0	1405	0	41	100	63
394	11	0	72	0	461	11	11	39	0	528	44	55	0	0	1555	0	25	39	0

1565	0	39	50	0
1575	0	48	69	0
1585	0	55	78	0
1595	0	59	100	11
1605	0	55	100	37
1615	0	55	100	44
1625	0	37	41	0
1635	0	41	48	0
1645	0	48	59	0
1655	0	63	78	0
1665	0	66	100	0
1675	0	66	100	37
1685	0	66	100	48
1765	0	44	29	0
1775	0	50	37	0
1785	0	66	50	0
1788	0	78	81	0
1795	0	84	100	0
1805	0	81	100	29
1815	0	81	100	50
2563	34	44	0	0
2573	44	63	0	0
2583	55	78	0	0
2593	63	100	0	0
2603	63	100	0	11
2613	63	100	0	18
2623	63	100	0	37
2567	39	44	0	0
2577	50	59	0	0
2587	59	74	0	0
2597	69	100	0	0
2607	69	100	0	15
2617	69	100	0	21
2627	66	100	0	29
2635	37	34	0	0
2645	44	41	0	0
2655	55	50	0	0
2665	66	72	0	0
2685	78	100	0	0
2695	78	100	0	41
2705	44	37	0	0
2715	59	48	0	0
2725	69	66	0	0
2735	78	100	0	0
2745	81	100	0	0
2755	81	100	0	0
2765	81	100	0	39
2975	39	0	11	0
2985	63	0	11	0
2995	75	15	11	0
3005	100	25	0	11
3015	100	18	0	25
3025	100	0	11	50
3035	100	0	18	63
3105	39	0	18	0
3115	55	0	25	0
3125	66	0	29	0
3135	81	0	29	0
3145	81	0	34	15
3155	84	0	37	37
3165	100	0	41	44
3242	39	0	25	0
3252	50	0	34	0
3262	59	0	41	0
3272	75	0	48	0
3282	84	0	55	0
3292	84	0	59	41
3302	100	0	59	50
3245	37	0	25	0
3255	48	0	37	0
3265	55	0	44	0
3275	72	0	50	0
3285	74	0	55	0
3295	81	0	63	0
3305	78	0	66	44
3248	37	0	29	0
3258	50	0	39	0
3268	66	0	50	0
3278	84	0	63	0
3288	100	0	66	0
3298	84	0	63	25
3308	100	0	63	59
3375	34	0	29	0
3385	44	0	37	0
3395	50	0	44	0
3405	63	0	59	0
3415	78	0	74	15
3425	78	0	74	39
3435	84	0	78	50
3935	0	0	66	0
3945	0	0	81	0
3955	11	0	100	0
3965	15	0	100	0
3975	0	0	100	29
3985	0	0	100	44
3995	0	0	100	63
4485	0	29	100	69
4495	0	25	81	50
4505	0	21	72	41
4515	0	15	48	29
4525	0	11	41	25
4535	0	0	34	18
4545	0	0	21	11
4625	0	50	100	75
4635	0	48	84	48
4645	0	39	66	37
4655	0	29	44	25
4665	0	25	37	18
4675	0	18	25	11
4685	0	11	21	11
4695	0	74	100	69
4705	0	59	69	50
4715	0	41	44	39
4725	0	37	39	34
4735	0	29	29	21
4745	0	25	25	18
4755	0	18	18	11
4975	0	72	75	74
4985	0	59	50	48
4995	0	48	41	39
5005	0	41	34	29
5015	0	34	21	18
5025	0	25	15	15
5035	0	18	11	11
5115	66	100	72	0
5125	63	81	55	0
5135	50	63	37	0
5145	34	44	18	0
5155	25	37	15	0
5165	18	29	11	0
5175	15	21	11	0
5185	78	100	100	0
5195	72	100	78	0
5205	55	59	50	0
5215	44	44	37	0
5225	37	39	29	0
5235	18	25	15	0
5245	15	18	11	0
5255	75	78	0	72
5265	74	69	0	50
5275	66	55	0	41
5285	48	39	0	34
5295	37	29	0	18
5305	25	21	0	15
5315	11	11	0	11
5395	74	41	0	72
5405	59	15	0	63
5415	48	11	0	50
5425	39	0	0	41
5435	25	0	0	34
5445	18	0	0	25
5455	11	0	0	15
5463	100	0	34	72
5473	66	0	37	55
5483	50	0	29	39
5493	37	0	21	29
5503	29	0	18	25
5513	21	0	11	11
5523	18	0	11	11
5467	78	0	41	78
5477	55	0	37	69
5487	39	0	25	55
5497	25	0	15	41
5507	18	0	11	37
5517	15	0	11	25
5527	11	0	11	18
5535	75	0	72	66
5545	55	0	50	50
5555	44	0	39	41
5565	37	0	34	34
5575	25	0	21	25
5585	18	0	18	18
5595	11	0	15	15
5605	69	0	63	78
5615	44	0	50	63
5625	37	0	37	50
5635	25	0	25	41
5645	18	0	21	34
5655	11	0	15	25
5665	11	0	11	18
5743	39	0	69	75
5753	34	0	74	63
5763	25	0	69	55
5773	15	0	44	41
5783	11	0	34	34
5793	11	0	29	25
5803	0	0	18	21
5747	39	0	100	72
5757	34	0	78	50
5767	21	0	63	41
5777	15	0	48	34
5787	11	0	37	18
5797	11	0	29	15
5807	0	0	18	15
5815	0	0	81	74
5825	0	0	78	59
5835	0	0	63	44
5845	0	0	48	37
5855	0	0	37	25
5865	0	0	29	21
5875	0	0	25	15

White	0	0	0	0
Yellow (Pantone 102)				
	0	0	100	0
Warm Red (Pantone 179)				
	0	74	81	0
Rubine Red				
	0	100	21	0
Rhodamine (Pantone 226)				
	0	100	0	0
Purple	44	81	0	0
Violet	75	84	0	0
Reflex Blue				
	100	72	0	11
Process Blue				
	100	0	0	0
Green	74	0	63	0