Subject: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Fri, 26 Apr 2002 21:17:58 GMT
View Forum Message <> Reply to Message

I took David's advice and used the subject heading when creating gifs from my display, this fixed the problem that the white background would always show up slightly grey. However, as I posted afterwards, the option reduces the number of colors by a factor of 2 roughly. It also makes the plots look 'grainy', which sounds like the same problem actually. Is there a way around this, i.e. I'd like white background when I ask for it, but I don't really want to lose the color resolution either. hmmmph!

as an aside, I'm trying to give people graphics that would insert in Word documents easily. Does IDL for Windows have a vector based image format (like WMF, EMF) that I can write to? My IDL 5.2 for Linux doesn't have (or appear to) anything but CGM.

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by Mark Hadfield on Mon, 29 Apr 2002 00:41:23 GMT
View Forum Message <> Reply to Message

"Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message news:slrnacjgsk.30q.nobody@pooh.nrel.gov...
> as an aside, I'm trying to give people graphics that would insert in
> Word documents easily. Does IDL for Windows have a vector based
> image format (like WMF, EMF) that I can write to? My IDL 5.2 for
> Linux doesn't have (or appear to) anything but CGM.

Am I correct in deducing that you don't consider it to be easy to insert Postscript into Word documents?

I use IDL 5.5 on Windows, which supports output to WMF, but when I want to insert a graphic into a Word document, my preferred format is EPS with a TIFF preview. As I'm sure you're aware this gives very good output on a Postscript printer and it's possible to make the preview resolution high enough for the figure to be recognisable on the screen.

--
Mark Hadfield "Ka puwaha et tai nei, Hoea tatou"
m.hadfield@niwa.co.nz

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by David Fanning on Mon, 29 Apr 2002 00:56:02 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

> I use IDL 5.5 on Windows, which supports output to WMF, but when I
> want to insert a graphic into a Word document, my preferred format is
> EPS with a TIFF preview. As I'm sure you're aware this gives very good
> output on a Postscript printer and it's possible to make the preview
> resolution high enough for the figure to be recognisable on the
> screen.

I use encapsulated PostScript files exclusively for IDL
output in my book. I always add a Preview (if I need one)
in GhostView, since I find IDL's preview capability ... uh,
anemic.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Mon, 29 Apr 2002 16:33:17 GMT
View Forum Message <> Reply to Message

It's not that I consider it difficult, it just isn't very easy to exchange
these types of files. Most of the authors I would give these figures to may
not appreciate the fact that though the screen image looks _bad_, the printed
output will look better. A lot of conference proceedings now use a web-sub-
mission process that converts the file to pdf, since I'm not the one who
will submit the document, I'm concerned that a .ps figure would not fare well
through this process.

On Mon, 29 Apr 2002 12:41:23 +1200, Mark Hadfield <m.hadfield@niwa.co.nz> wrote:
> "Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message

> news:slrnacjgsk.30q.nobody@pooh.nrel.gov...
>> as an aside, I'm trying to give people graphics that would insert in
>> Word documents easily. Does IDL for Windows have a vector based
>> image format (like WMF, EMF) that I can write to? My IDL 5.2 for
>> Linux doesn't have (or appear to) anything but CGM.
>
> Am I correct in deducing that you don't consider it to be easy to
> insert Postscript into Word documents?
>
> I use IDL 5.5 on Windows, which supports output to WMF, but when I
> want to insert a graphic into a Word document, my preferred format is
> EPS with a TIFF preview. As I'm sure you're aware this gives very good
> output on a Postscript printer and it's possible to make the preview
> resolution high enough for the figure to be recognisable on the
> screen.
>
> --
> Mark Hadfield "Ka puwaha et tai nei, Hoea tatou"
> m.hadfield@niwa.co.nz
> National Institute for Water and Atmospheric Research (NIWA)
>
>

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by David Fanning on Tue, 30 Apr 2002 02:15:35 GMT
View Forum Message <> Reply to Message

Steve Smith<steven_smith> (nobody@nowhere.com) writes:

> It's not that I consider it difficult, it just isn't very easy to exchange
> these types of files. Most of the authors I would give these figures to may
> not appreciate the fact that though the screen image looks _bad_, the printed
> output will look better. A lot of conference proceedings now use a web-sub-
> mission process that converts the file to pdf, since I'm not the one who
> will submit the document, I'm concerned that a .ps figure would not fare well
> through this process.

On the contrary, it is always those *other* image
types that give the PDF Distiller fits. Those Adobe
guys know PostScript. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Tue, 30 Apr 2002 15:02:52 GMT
View Forum Message <> Reply to Message

On Mon, 29 Apr 2002 20:15:35 -0600, David Fanning <david@dfanning.com> wrote:
> Steve Smith<steven_smith> (nobody@nowhere.com) writes:
>
>> It's not that I consider it difficult, it just isn't very easy to exchange
>> these types of files. Most of the authors I would give these figures to may
>> not appreciate the fact that though the screen image looks _bad_, the printed
>> output will look better. A lot of conference proceedings now use a web-sub-
>> mission process that converts the file to pdf, since I'm not the one who
>> will submit the document, I'm concerned that a .ps figure would not fare well
>> through this process.
>
> On the contrary, it is always those *other* image
> types that give the PDF Distiller fits. Those Adobe
> guys know PostScript. :-)
>
> Cheers,
>
> David

Hi David-
Oh, I believe that Adobe should know postscript (as I think they were one of
the principal developers of Postscript), it's just that fear of the unknown.
The PC version of Word always shows PS as a blank box (the Mac Version could
handle it), and I've always been uncomfortable with that. I know if you add the
preview, it will show a picture, but can this be scaled within word similar to
other figures transferred into Word via vector formats like WMF (which I
think is what comes off WinXX's clipboard)? If I make a figure, give it to a
co-author, they may want to re-size it when they format their document. They're
not going to know or much less care about the subtleties of document formats,
fonts or printing. It's a WYSIWYG world now, not like when I was in grad
school: we used IDL for all postscript output and inserted this into LaTeX

documents. The output was beautiful, far exceeding what could be done with Word
and a PC at the time. But I don't run in to too many who know, much less use
TeX anymore :P !

>
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by David Fanning on Tue, 30 Apr 2002 15:35:16 GMT
View Forum Message <> Reply to Message

Steve Smith<steven_smith> (nobody@nowhere.com) writes:

> Oh, I believe that Adobe should know postscript (as I think they were one of
> the principal developers of Postscript), it's just that fear of the unknown.
> The PC version of Word always shows PS as a blank box (the Mac Version could
> handle it), and I've always been uncomfortable with that. I know if you add the
> preview, it will show a picture, but can this be scaled within word similar to
> other figures transferred into Word via vector formats like WMF (which I
> _think_ is what comes off WinXX's clipboard)?

I just tried a fairly complicated scene with a map, image on map,
colorbar, etc. Made a PS file in IDL. Converted that to an EPS
file in Ghostview with several different types of preview images
(TIFF, Windows metafile, etc.). All behaved perfectly in Windows,
allowing scaling, etc. Preview looked like, well... a preview. But
all printed nicely.

Interestingly, if I made an EPS file in IDL, then the bounding
box was made too small and some of the text of my colorbar was
cut off. (This happened in IDL 5.4 and 5.5. Is this a bug?)
But scaling, printing, etc. were all as I expected.

> If I make a figure, give it to a
> co-author, they may want to re-size it when they format their document. They're

> not going to know or much less care about the subtleties of document formats,
> fonts or printing. It's a WYSIWYG world now

I think even the older professors would be able
to negotiate these EPS files. :-)

> , not like when I was in grad
> school: we used IDL for all postscript output and inserted this into LaTeX
> documents. The output was beautiful, far exceeding what could be done with Word
> and a PC at the time. But I don't run in to too many who know, much less use
> TeX anymore :P !

A pity, because TeX was something that separated the programmers from
the wantabees. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Tue, 30 Apr 2002 18:37:58 GMT
View Forum Message <> Reply to Message

On Tue, 30 Apr 2002 09:35:16 -0600, David Fanning <david@dfanning.com> wrote:
> I just tried a fairly complicated scene with a map, image on map,
> colorbar, etc. Made a PS file in IDL. Converted that to an EPS
> file in Ghostview with several different types of preview images
> (TIFF, Windows metafile, etc.). All behaved perfectly in Windows,
> allowing scaling, etc. Preview looked like, well... a preview. But
> all printed nicely.
>
> Interestingly, if I made an EPS file in IDL, then the bounding
> box was made too small and some of the text of my colorbar was
> cut off. (This happened in IDL 5.4 and 5.5. Is this a bug?)
> But scaling, printing, etc. were all as I expected.
>

I ended up making PS and using gsview to make EMF files on a Windows98
machine. I'll have to go back and make some EPS files w/preview and
compare the two. Incidentally, it's usually the text that is spoiled in
the conversions.

<snip>

>> and a PC at the time. But I don't run in to too many who know, much less use
>> TeX anymore :P !
>
> A pity, because TeX was something that separated the programmers from
> the wantabees. :-)
>

let's hope history is not repeated with IDL! ('naw ... never happen ;-) !).

> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Tue, 30 Apr 2002 22:10:28 GMT
View Forum Message <> Reply to Message

On Tue, 30 Apr 2002 18:37:58 -0000, Steve Smith<steven_smith>
<nobody@nowhere.com> wrote:
> On Tue, 30 Apr 2002 09:35:16 -0600, David Fanning <david@dfanning.com> wrote:
>> I just tried a fairly complicated scene with a map, image on map,
>> colorbar, etc. Made a PS file in IDL. Converted that to an EPS
>> file in Ghostview with several different types of preview images
>> (TIFF, Windows metafile, etc.). All behaved perfectly in Windows,
>> allowing scaling, etc. Preview looked like, well... a preview. But
>> all printed nicely.
>>
>> Interestingly, if I made an EPS file in IDL, then the bounding
>> box was made too small and some of the text of my colorbar was
>> cut off. (This happened in IDL 5.4 and 5.5. Is this a bug?)
>> But scaling, printing, etc. were all as I expected.

>>
>
> I ended up making PS and using gsview to make EMF files on a Windows98
> machine. I'll have to go back and make some EPS files w/preview and
> compare the two. Incidentally, it's usually the text that is spoiled in
> the conversions.

< snip$^{\infty}$ >

I tried it (making EPS w/preview --> Word, call it CASE1), printing looks beautiful (and better quality than the EMF images inserted into word, call that CASE2), but the screen is ugly :((CASE2 looks better on the screen). The most dissappointing part is that when printing the documents to a PDF file (with Adobe PDFwriter), the EPS/w preview images (CASE1) are inferior to the EMF images (CASE2) in the final PDF document >:| !!! Worse yet, the trend is preserved when printing the PDF's. So maybe the postscript is NOT used by the PDF writer?

I'm speculating that perhaps the thing to do is create a postscript document by printing the Word document to a file (and massaging out the non-postscript commands WinXX sticks in) and converiting that directly to PDF. Unfortunately, it doesn't fix the original problem: I need to give a figure that can be inserted into Word by a colleauge.

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by Mark Hadfield on Tue, 30 Apr 2002 23:44:52 GMT
View Forum Message <> Reply to Message

"Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message news:slrnacu5e9.e6b.nobody@pooh.nrel.gov...

> I'm speculating that perhaps the thing to do is create a postscript
> document by printing the Word document to a file (and massaging out
> the non-postscript commands WinXX sticks in) and converiting that
> directly to PDF.

Yes. I tried this yesterday and confirmed that this works. I created an IDL EPS file, added a preview with GSview, imported it into Word, printed the Word document to a file via a Windows Postscript printer driver and then processed the resulting PS file using an automated

convertor we have here. (It uses Adobe Distiller, I think.) No need to massage the PS file. Result: PDF document with nice vector diagram.

> Unfortunately, it doesn't fix the original problem: I need to give a
> figure that can be inserted into Word by a colleauge.

Hhmmm. Yes, this is a problem, Some possibilities:

- I used to have a third-party Word graphics filter that imported Postscript in vector form. It's no longer available, however, and your colleague is unlikely to have it installed in any case.

- Pstoedit can convert EPS to WMF or EMF. Unfortunately it doesn't handle Postscript "image" elements, so it is of limited use.

= Corel Draw? Adobe Illustrator? I think both of these can import EPS and export in various Word-friendly vector formats. They cost serious money, though.

- When I flirted with Linux a few months ago I noticed that Linux word processors could import EPS files. (This is a mixed blessing, I might add, as complicated diagrams can slow down display badly.) Perhaps there's one that imports EPS and exports Word?

Good luck.

--
Mark Hadfield "Ka puwaha et tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by paul.krummel on Wed, 01 May 2002 02:46:34 GMT
View Forum Message <> Reply to Message

nobody@nowhere.com (Steve Smith<steven_smith>) wrote in message news:<slrnacu5e9.e6b.nobody@pooh.nrel.gov>...
> On Tue, 30 Apr 2002 18:37:58 -0000, Steve Smith<steven_smith>
> <nobody@nowhere.com> wrote:
>
> <...snip...>
>
> I'm speculating that perhaps the thing to do is create a postscript document by
> printing the Word document to a file (and massaging out the non-postscript
> commands WinXX sticks in) and converiting that directly to PDF. Unfortunately,
> it doesn't fix the original problem: I need to give a figure that can be
> inserted into Word by a colleauge.

I thought I would just add what I do as I come across this all the time while trying to share files or show some results. The following is all done on a Windows 2000 PC.

As has already been said by a few people, if I am producing a document that is to be printed on a PostScript printer, then I use PostScript from IDL, turn it to an EPS and add a preview with GSView. I then import this into Word and I can rescale it etc and all prints very nicely.

However, if I am giving a presentation with say PowerPoint, then I like to use images instead of EPS files. I have been using PNG files with great success. I usually convert my PS output to 300dpi PNG images using gsview. These import nicely into Word or Powerpoint and at this resolution look very good on screen and even when printed. In fact, this is how I give plots and figures to other people in our group here as they can easily use them. These images scale well in Word or Powerpoint, and they have even been used in large posters which look good printed.

I got sick of doing all this manually, so I now incorporate the conversion to PNG and cropping the image into some of my IDL code. Here is a simple example that makes a simple plot in postscript, spawns gswin32c to convert to 300dpi PNG file, then calls a routine I wrote called 'crop_image.pro' to automatically crop the image.

Anyway, for what it is worth, this is what I do!
Cheers, Paul

```
pro ps_image_crop
;
; Quick routine to make a PS file,
; convert it to PNG and crop the
; image.
; PBK 1 May 2002.
; ++++
; Setup a test PS file
ps_file='c:\krum\gaslab\muck\test.ps'
png_file='c:\krum\gaslab\muck\test.png'
png_crop='c:\krum\gaslab\muck\crop.png'
;
; Create some data
x=findgen(200)*0.1
y=sin(x)
;
```

```
; ++++
; Setup plotting
!p.multi=0
set_plot,'ps'
device,file=ps_file,/helvetica,/color,bits=8
!p.font=0
!p.thick=4
!x.thick=4
!y.thick=4
;
; load rainbow colour table
loadct,13
;
plot, x, y, xtitle='X', ytitle='sin(X)', $
 title='TEST2 - abc ijk xyz',/nodata
oplot,x,y,color=255
oplot,x,y,color=64,psym=1
;
; close ps device
device,/close
;
; ++++
; convert the ps file into PNG files
spawn,'"c:\program files\gs\gs7.04\bin\gswin32c"'+ $
 ' -sDEVICE=png256 -r300 -q -dNOPAUSE -dBATCH '+ $
 '-sPAPERSIZE=a4 -sOutputFile='+png_file+' '+ps_file, $
 /log_output
;
; Note version of gostscript and path may need to be
; changed depending on what you have installed.
; Note usage of 'gswin32c' which is the command line
; version of gswin32. Here is a quick explanation
; of some of the command line options:
; -sDEVICE=png256
; convert to PNG with 256 colours or 16 million (16m)
;
; -r300  -> convert at 300 dpi see documentation for
; available resolutions.
;
; -q -dNOPAUSE -dBATCH
; various options for batch processing and being quiet
;
; -sPAPERSIZE=a4  -> paper size
;
; -sOutputFile=png_file
; output file (note for multiple pages you can set
; something like this for the file name 'out_page%d.png'
; where %d will be the page number, so if there are
```

```
; 4 pages in the PS file then there will be 4 image files
; called out_page1.png, out_page2.png, out_page3.png
; and out_page4.png.
;
; ++++
; Now crop the image automatically with a 2% buffer around it
; and write out to new file.
crop_image,png_file,out_file=png_crop
; or crop it and overwrite the file
;crop_image,png_file
; or if in landscape, crop it and rotate it
;crop_image,png_file,rot=1
;
; ++++
;
beep
end

****************************************
;+
; NAME:
; CROP_IMAGE
;
; PURPOSE:
; This procedure performs an autocrop on an image in
; IDL based on white space or white background. It
; will work with 8-bit (256 colours) or 24-bit (true
; colour) images. It can also rotate images if required.
;
; CATEGORY:
; Image manipulation
;
; CALLING SEQUENCE:
; CROP_IMAGE, Image_File, OUT_FILE=Out_File, COORDS=CoOrds, ROT=Rot
;
; INPUTS:
; Image_File: This is the path and filename of the image
;   file to crop. Must be a string.
;
; KEYWORD PARAMETERS:
; OUT_FILE: Set this keyword to the desired output filename. If
;   this keyword is not set then the default is to overwrite
;   the input image file (Image_File above) with the cropped
;   image. Currently the output image will be in the same
;   format as the input image. Must be a string.
;
; COORDS:  Set this keyword to a four element array containing
;   the min and max pixel locations where you want the image
```

```
;   to be cropped eg. [x_min,y_min,x_max,y_max]. If this
;   keyword is not set, the default is to crop the image to
;   allow a 2% buffer of the total image pixel size on all
;   sides of the image from the first non white pixels.
;   NOTE: In some image formats the first pixel row starts at
;   the top of the image while in others it is at the bottom
;   of the image.
;
; ROT:  Set this keyword to the rotation (direction) that
;   is required based on the following table:
;  Direction  Transpose?  Rotation Counterclockwise  X1  Y1
;   0          No        None                 X0 Y0
;   1          No         90ï¿½                 -Y0  X0
;   2          No        180ï¿½                 -X0  -Y0
;   3          No        270ï¿½                 Y0  -X0
;   4          Yes       None                 Y0  X0
;   5          Yes        90ï¿½                 -X0  Y0
;   6          Yes       180ï¿½                 -Y0  -X0
;   7          Yes       270ï¿½                 X0  -Y0
;
;
; OUTPUTS:
; Writes an image to file, see OUT_FILE above.
;
; RESTRICTIONS:
; Only works for 8- or 24-bit images. Requires IDL 5.4 or higher.
;
; PROCEDURE:
; Straightforward image reading and cropping (sub array extraction)
; based on COORDS OR simple checking for first non-white pixel
; locations (if COORDS keyword not set) and taking 2% buffers
; around these.
;
; EXAMPLE:
; Read in the image CapeGrim_Map.png, auto crop it and write out
; to same file name:
;  IDL> crop_image,'c:\krum\gaslab\map\CapeGrim_Map.png'
; OR also rotate it 90 degress counter clockwise
;  IDL> crop_image,'c:\krum\gaslab\map\CapeGrim_Map.png', rot=1
;
; MODIFICATION HISTORY:
;  Written by: Paul Krummel, CSIRO Atmospheric Research, 28 Febuary 2001.
;  Modified by Paul Krummel, CAR, 25 April 2001. Added COORDS keyword.
; Modified by Paul Krummel, CAR, 23 August 2001. Added ROT keyword.
; Modified by Paul Krummel, CAR, 17 January 2002. Added file checking
;  to see if input image file variable (im_file) is of type string
;  and if input image file exists or not.
; Modified by Paul Krummel, CAR, 31 January 2002. Fixed oversight (bug)
```

```
;  with rotating the image when it is 24 bit!! ROT now works for
;  24-bit images.
;-

PRO CROP_IMAGE, Im_File, OUT_FILE=Out_File, COORDS=CoOrds, ROT=Rot
;
; =====>> HELP
;
on_error,2
if (N_PARAMS(0) NE 1) or keyword_set(help) then begin
   doc_library,'CROP_IMAGE'
   if N_PARAMS(0) NE 1 and not keyword_set(help) then $
            message,'Incorrect number of parameters, see above for usage.'
   return
endif
;
; ++++
; THINGS STILL TO DO:
; Add options for output image format
; Allow passing in an image array and passing image back again i.e.
; no reading/writing from file on disk.
;
; ++++
;
; Make sure input image file variable is a string
if size(im_file,/type) ne 7 then message,'"image_file" must be a string!!'
;
; Check that the image files exists and is readable!
if not file_test(im_file,/read) then message,'Bugger ... '+im_file+ $
     ' does not exist or is not readable!!'
;
; Read in the image
ok=query_image(im_file,info)
if not ok then message,'Oh BUGGER ... '+im_file+ $
   ' is not a proper or supported image file!'
image = READ_image ( im_File, R, G, B)
;
; ++++
; Check to see if coords were passed in, if so use them!
if n_elements(coords) gt 0 then begin
 c_min=coords[0] & c_max=coords[2]
 r_min=coords[1] & r_max=coords[3]
;
; ++++
; If not then autocrop
ENDIF ELSE BEGIN
;
; ++++
```

```
; Check if it is an 8 bit or 24 bit image
 case info.channels of
  1: BEGIN ; 8-bit
;
;   find the min colour used in the image
   i_min=min(image)
;
;    Check for white (r,g,b=255), just a test!
;   White is usually ALWAYS at 255 which usually is
;    r=255, g=255, b=255, but just check to make sure!
   rgb=fix(r)+fix(g)+fix(b)
   wh=where(rgb[i_min:255] eq 765)+i_min
;   white can occur more than once!
;   Is usually 255 though, just use the max
   wh=max(wh)
;   Find where image is not white
   not_white=where(image ne wh, cnt_nw)
;
    END
;
  3: BEGIN ; 24-bit
;
;   24 bit image -> 16.7 million colours
   rgb=total(image,1)
;   White is 765
   wh=765
;   Find where image is not white
   not_white=where(rgb ne wh, cnt_nw)
;
    END
 endcase
;
;  Convert this to columns and rows
 ncol = info.dimensions[0]
 nrow = info.dimensions[1]
 col = not_white MOD ncol
 row = not_white / ncol
;
;  Find the min and max of the col and row
 c_min=min(col,max=c_max)
 r_min=min(row,max=r_max)
;
;  Find average of col and row sizes and take 2% of it
 buff=ceil((((c_max-c_min)+(r_max-r_min))/2)*0.02)
;
;  Now add this to make buffer around image and find new dimensions
 c_min= c_min-buff > 0
 c_max= c_max+buff < ncol-1
```

```
 r_min= r_min-buff > 0
 r_max= r_max+buff < nrow-1
;
; ++++
ENDELSE
;
; ++++
; resize the image, with 'buff' pixels added on all sides
case info.channels of
 1: image=image[c_min:c_max,r_min:r_max] ; 8 bit
 3: image=image[*,c_min:c_max,r_min:r_max] ; 24 bit
endcase
;
; ++++
; if requested, rotate the image
if n_elements(rot) gt 0 then begin
 case info.channels of
  1: image=rotate(image,rot) ; 8 bit image
  3: begin   ; 24 bit image
;   Rotate the individual channels of the image by requested amount
   ir=rotate(reform(image[0,*,*]),rot) ; 24 bit red
   ig=rotate(reform(image[1,*,*]),rot) ; 24 bit green
   ib=rotate(reform(image[2,*,*]),rot) ; 24 bit blue
;          Reassign the individual channels to 3D and delete the variable
   sz=size(ir,/dimension)
   image=bytarr(3,sz[0],sz[1])
   image[0,*,*]=temporary(ir)
   image[1,*,*]=temporary(ig)
   image[2,*,*]=temporary(ib)
    end
 endcase
endif
;
; ++++
; Write out the image
if keyword_set(out_file) then o_file=out_file $
else o_file=im_file
WRITE_image, o_file, info.type, Image, R, G, B
;
; ++++
end
```

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by David Fanning on Wed, 01 May 2002 03:47:50 GMT
View Forum Message <> Reply to Message

Paul Krummel (paul.krummel@csiro.au) writes:

> I thought I would just add what I do as I come across this all
> the time while trying to share files or show some results. The
> following is all done on a Windows 2000 PC.

[2000+ lines of code deleted]

Who said it was hard to get high-quality output out of
IDL! :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Wed, 01 May 2002 18:03:54 GMT
View Forum Message <> Reply to Message

thank you Paul and Mark for your ideas. Yes, if I can get my hands on the final
document, I can always print to a file and convert to PDF and things look good.
The PNG's do look better than other bitmapped graphics like gif or tiff (I'm
curious why this is?), the figures I made are greyscale shaded surfaces. when
converted to png (png16m is ghostscript device) they look good, but a little
greyed (text is not sharp as it is in the vectorized image formats). It's too
bad there doesn't seem to exist any universal vector-based image file format.
A format that could be scaled, preserve fonts contain bitmap data and do it on
any platform (I know, you're saying Postscript!) AND be supported by the most
prevalent Word-processing program (was Word Perfect any better?). I use Linux
IDL, but I think your routine should work with gs on linux just as well. It
may be worth my while to try it out if this continues (the demand for Windows
insertable figures continues here).

On 30 Apr 2002 19:46:34 -0700, Paul Krummel <paul.krummel@csiro.au> wrote:
> nobody@nowhere.com (Steve Smith<steven_smith>) wrote in message
news:<slrnacu5e9.e6b.nobody@pooh.nrel.gov>...
>> On Tue, 30 Apr 2002 18:37:58 -0000, Steve Smith<steven_smith>
>> <nobody@nowhere.com> wrote:
>>
>> <...snip...>
>>

>> I'm speculating that perhaps the thing to do is create a postscript document by
>> printing the Word document to a file (and massaging out the non-postscript
>> commands WinXX sticks in) and converiting that directly to PDF. Unfortunately,
>> it doesn't fix the original problem: I need to give a figure that can be
>> inserted into Word by a colleauge.
>
> I thought I would just add what I do as I come across this all
> the time while trying to share files or show some results. The
> following is all done on a Windows 2000 PC.
>
> As has already been said by a few people, if I am producing
> a document that is to be printed on a PostScript printer,
> then I use PostScript from IDL, turn it to an EPS and add
> a preview with GSView. I then import this into Word and
> I can rescale it etc and all prints very nicely.
>
> However, if I am giving a presentation with say PowerPoint,
> then I like to use images instead of EPS files. I have been
> using PNG files with great success. I usually convert my PS
> output to 300dpi PNG images using gsview. These import nicely
> into Word or Powerpoint and at this resolution look very good
> on screen and even when printed. In fact, this is how I give
> plots and figures to other people in our group here as they
> can easily use them. These images scale well in Word or
> Powerpoint, and they have even been used in large posters
> which look good printed.
>
> I got sick of doing all this manually, so I now incorporate
> the conversion to PNG and cropping the image into some of
> my IDL code. Here is a simple example that makes a simple
> plot in postscript, spawns gswin32c to convert to 300dpi
> PNG file, then calls a routine I wrote called 'crop_image.pro'
> to automatically crop the image.
>
> Anyway, for what it is worth, this is what I do!
> Cheers, Paul
>
> pro ps_image_crop
> ;
> ; Quick routine to make a PS file,
> ; convert it to PNG and crop the
> ; image.
> ; PBK 1 May 2002.
> ; ++++
> ; Setup a test PS file
> ps_file='c:\krum\gaslab\muck\test.ps'
> png_file='c:\krum\gaslab\muck\test.png'
> png_crop='c:\krum\gaslab\muck\crop.png'

```
> ;
> ; Create some data
> x=findgen(200)*0.1
> y=sin(x)
> ;
> ; ++++
> ; Setup plotting
> !p.multi=0
> set_plot,'ps'
> device,file=ps_file,/helvetica,/color,bits=8
> !p.font=0
> !p.thick=4
> !x.thick=4
> !y.thick=4
> ;
> ; load rainbow colour table
> loadct,13
> ;
> plot, x, y, xtitle='X', ytitle='sin(X)', $
>  title='TEST2 - abc ijk xyz',/nodata
> oplot,x,y,color=255
> oplot,x,y,color=64,psym=1
> ;
> ; close ps device
> device,/close
> ;
> ; ++++
> ; convert the ps file into PNG files
> spawn,'"c:\program files\gs\gs7.04\bin\gswin32c"'+ $
>  ' -sDEVICE=png256 -r300 -q -dNOPAUSE -dBATCH '+ $
>  '-sPAPERSIZE=a4 -sOutputFile='+png_file+' '+ps_file, $
>  /log_output
> ;
> ; Note version of gostscript and path may need to be
> ; changed depending on what you have installed.
> ; Note usage of 'gswin32c' which is the command line
> ; version of gswin32. Here is a quick explanation
> ; of some of the command line options:
> ; -sDEVICE=png256
> ; convert to PNG with 256 colours or 16 million (16m)
> ;
> ; -r300  -> convert at 300 dpi see documentation for
> ; available resolutions.
> ;
> ; -q -dNOPAUSE -dBATCH
> ; various options for batch processing and being quiet
> ;
> ; -sPAPERSIZE=a4  -> paper size
```

> ;
> ; -sOutputFile=png_file
> ; output file (note for multiple pages you can set
> ; something like this for the file name 'out_page%d.png'
> ; where %d will be the page number, so if there are
> ; 4 pages in the PS file then there will be 4 image files
> ; called out_page1.png, out_page2.png, out_page3.png
> ; and out_page4.png.
> ;
> ; ++++
> ; Now crop the image automatically with a 2% buffer around it
> ; and write out to new file.
> crop_image,png_file,out_file=png_crop
> ; or crop it and overwrite the file
> ;crop_image,png_file
> ; or if in landscape, crop it and rotate it
> ;crop_image,png_file,rot=1
> ;
> ; ++++
> ;
> beep
> end
>
> **
> ;+
> ; NAME:
> ; CROP_IMAGE
> ;
> ; PURPOSE:
> ; This procedure performs an autocrop on an image in
> ; IDL based on white space or white background. It
> ; will work with 8-bit (256 colours) or 24-bit (true
> ; colour) images. It can also rotate images if required.
> ;
> ; CATEGORY:
> ; Image manipulation
> ;
> ; CALLING SEQUENCE:
> ; CROP_IMAGE, Image_File, OUT_FILE=Out_File, COORDS=CoOrds, ROT=Rot
> ;
> ; INPUTS:
> ; Image_File: This is the path and filename of the image
> ;   file to crop. Must be a string.
> ;
> ; KEYWORD PARAMETERS:
> ; OUT_FILE: Set this keyword to the desired output filename. If
> ;   this keyword is not set then the default is to overwrite
> ;   the input image file (Image_File above) with the cropped

> ;   image. Currently the output image will be in the same
> ;   format as the input image. Must be a string.
> ;
> ; COORDS: Set this keyword to a four element array containing
> ;   the min and max pixel locations where you want the image
> ;   to be cropped eg. [x_min,y_min,x_max,y_max]. If this
> ;   keyword is not set, the default is to crop the image to
> ;   allow a 2% buffer of the total image pixel size on all
> ;   sides of the image from the first non white pixels.
> ;   NOTE: In some image formats the first pixel row starts at
> ;   the top of the image while in others it is at the bottom
> ;   of the image.
> ;
> ; ROT: Set this keyword to the rotation (direction) that
> ;   is required based on the following table:
> ; Direction Transpose? Rotation Counterclockwise X1 Y1
> ;   0   No   None   X0 Y0
> ;   1   No   90ï¿½   -Y0 X0
> ;   2   No   180ï¿½   -X0 -Y0
> ;   3   No   270ï¿½   Y0 -X0
> ;   4   Yes   None   Y0 X0
> ;   5   Yes   90ï¿½   -X0 Y0
> ;   6   Yes   180ï¿½   -Y0 -X0
> ;   7   Yes   270ï¿½   X0 -Y0
> ;
> ;
> ; OUTPUTS:
> ; Writes an image to file, see OUT_FILE above.
> ;
> ; RESTRICTIONS:
> ; Only works for 8- or 24-bit images. Requires IDL 5.4 or higher.
> ;
> ; PROCEDURE:
> ; Straightforward image reading and cropping (sub array extraction)
> ; based on COORDS OR simple checking for first non-white pixel
> ; locations (if COORDS keyword not set) and taking 2% buffers
> ; around these.
> ;
> ; EXAMPLE:
> ; Read in the image CapeGrim_Map.png, auto crop it and write out
> ; to same file name:
> ; IDL> crop_image,'c:\krum\gaslab\map\CapeGrim_Map.png'
> ; OR also rotate it 90 degress counter clockwise
> ; IDL> crop_image,'c:\krum\gaslab\map\CapeGrim_Map.png', rot=1
> ;
> ; MODIFICATION HISTORY:
> ; Written by: Paul Krummel, CSIRO Atmospheric Research, 28 Febuary 2001.
> ; Modified by Paul Krummel, CAR, 25 April 2001. Added COORDS keyword.

```
> ; Modified by Paul Krummel, CAR, 23 August 2001. Added ROT keyword.
> ; Modified by Paul Krummel, CAR, 17 January 2002. Added file checking
> ;  to see if input image file variable (im_file) is of type string
> ;  and if input image file exists or not.
> ; Modified by Paul Krummel, CAR, 31 January 2002. Fixed oversight (bug)
> ;  with rotating the image when it is 24 bit!! ROT now works for
> ;  24-bit images.
> ;-
>
> PRO CROP_IMAGE, Im_File, OUT_FILE=Out_File, COORDS=CoOrds, ROT=Rot
> ;
> ; =====>> HELP
> ;
> on_error,2
> if (N_PARAMS(0) NE 1) or keyword_set(help) then begin
>    doc_library,'CROP_IMAGE'
>    if N_PARAMS(0) NE 1 and not keyword_set(help) then $
>             message,'Incorrect number of parameters, see above for usage.'
>    return
> endif
> ;
> ; ++++
> ; THINGS STILL TO DO:
> ; Add options for output image format
> ; Allow passing in an image array and passing image back again i.e.
> ; no reading/writing from file on disk.
> ;
> ; ++++
> ;
> ; Make sure input image file variable is a string
> if size(im_file,/type) ne 7 then message,'"image_file" must be a string!!'
> ;
> ; Check that the image files exists and is readable!
> if not file_test(im_file,/read) then message,'Bugger ... '+im_file+ $
>      ' does not exist or is not readable!!'
> ;
> ; Read in the image
> ok=query_image(im_file,info)
> if not ok then message,'Oh BUGGER ... '+im_file+ $
>    ' is not a proper or supported image file!'
> image = READ_image ( im_File, R, G, B)
> ;
> ; ++++
> ; Check to see if coords were passed in, if so use them!
> if n_elements(coords) gt 0 then begin
>  c_min=coords[0] & c_max=coords[2]
>  r_min=coords[1] & r_max=coords[3]
> ;
```

```
> ; ++++
> ; If not then autocrop
> ENDIF ELSE BEGIN
> ;
> ; ++++
> ; Check if it is an 8 bit or 24 bit image
>  case info.channels of
>   1: BEGIN ; 8-bit
> ;
> ;   find the min colour used in the image
>    i_min=min(image)
> ;
> ;    Check for white (r,g,b=255), just a test!
> ;   White is usually ALWAYS at 255 which usually is
> ;    r=255, g=255, b=255, but just check to make sure!
>    rgb=fix(r)+fix(g)+fix(b)
>    wh=where(rgb[i_min:255] eq 765)+i_min
> ;   white can occur more than once!
> ;   Is usually 255 though, just use the max
>    wh=max(wh)
> ;   Find where image is not white
>    not_white=where(image ne wh, cnt_nw)
> ;
>     END
> ;
>   3: BEGIN ; 24-bit
> ;
> ;   24 bit image -> 16.7 million colours
>    rgb=total(image,1)
> ;   White is 765
>    wh=765
> ;   Find where image is not white
>    not_white=where(rgb ne wh, cnt_nw)
> ;
>     END
>  endcase
> ;
> ;  Convert this to columns and rows
>  ncol = info.dimensions[0]
>  nrow = info.dimensions[1]
>  col = not_white MOD ncol
>  row = not_white / ncol
> ;
> ;  Find the min and max of the col and row
>  c_min=min(col,max=c_max)
>  r_min=min(row,max=r_max)
> ;
> ;  Find average of col and row sizes and take 2% of it
```

```
>  buff=ceil((((c_max-c_min)+(r_max-r_min))/2)*0.02)
> ;
> ;  Now add this to make buffer around image and find new dimensions
>  c_min= c_min-buff > 0
>  c_max= c_max+buff < ncol-1
>  r_min= r_min-buff > 0
>  r_max= r_max+buff < nrow-1
> ;
> ; ++++
> ENDELSE
> ;
> ; ++++
> ; resize the image, with 'buff' pixels added on all sides
> case info.channels of
>  1: image=image[c_min:c_max,r_min:r_max] ; 8 bit
>  3: image=image[*,c_min:c_max,r_min:r_max] ; 24 bit
> endcase
> ;
> ; ++++
> ; if requested, rotate the image
> if n_elements(rot) gt 0 then begin
>  case info.channels of
>   1: image=rotate(image,rot) ; 8 bit image
>   3: begin   ; 24 bit image
> ;   Rotate the individual channels of the image by requested amount
>    ir=rotate(reform(image[0,*,*]),rot) ; 24 bit red
>    ig=rotate(reform(image[1,*,*]),rot) ; 24 bit green
>    ib=rotate(reform(image[2,*,*]),rot) ; 24 bit blue
> ;          Reassign the individual channels to 3D and delete the variable
>    sz=size(ir,/dimension)
>    image=bytarr(3,sz[0],sz[1])
>    image[0,*,*]=temporary(ir)
>    image[1,*,*]=temporary(ig)
>    image[2,*,*]=temporary(ib)
>     end
>  endcase
> endif
> ;
> ; ++++
> ; Write out the image
> if keyword_set(out_file) then o_file=out_file $
> else o_file=im_file
> WRITE_image, o_file, info.type, Image, R, G, B
> ;
> ; ++++
> end
```

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: color_quan(...., Cube=6) makes white white, but ...
Posted by nobody@nowhere.com (S on Wed, 01 May 2002 18:13:13 GMT
View Forum Message <> Reply to Message

thanks for your help, Mark. I have Corel Draw, it makes EMF that look identical in quality to the one's ghostscript makes, so there's no need to use it (or in general, buy it for that purpose alone). I think between your comments and others, I've pretty much put a nail in this coffin. I really appreciate the help, thanks.

On Wed, 1 May 2002 11:44:52 +1200, Mark Hadfield <m.hadfield@niwa.co.nz> wrote:
> "Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message
> news:slrnacu5e9.e6b.nobody@pooh.nrel.gov...
>
>> I'm speculating that perhaps the thing to do is create a postscript
>> document by printing the Word document to a file (and massaging out
>> the non-postscript commands WinXX sticks in) and converiting that
>> directly to PDF.
>
> Yes. I tried this yesterday and confirmed that this works. I created
> an IDL EPS file, added a preview with GSview, imported it into Word,
> printed the Word document to a file via a Windows Postscript printer
> driver and then processed the resulting PS file using an automated
> convertor we have here. (It uses Adobe Distiller, I think.) No need to
> massage the PS file. Result: PDF document with nice vector diagram.
>
>> Unfortunately, it doesn't fix the original problem: I need to give a
>> figure that can be inserted into Word by a colleauge.
>
> Hhmmm. Yes, this is a problem, Some possibilities:
>
> - I used to have a third-party Word graphics filter that imported
> Postscript in vector form. It's no longer available, however, and
> your colleague is unlikely to have it installed in any case.
>
> - Pstoedit can convert EPS to WMF or EMF. Unfortunately it doesn't
> handle Postscript "image" elements, so it is of limited use.
>
> = Corel Draw? Adobe Illustrator? I think both of these can import
> EPS and export in various Word-friendly vector formats. They cost
> serious money, though.

>
> - When I flirted with Linux a few months ago I noticed that Linux
> word processors could import EPS files. (This is a mixed blessing, I
> might add, as complicated diagrams can slow down display badly.)
> Perhaps there's one that imports EPS and exports Word?
>
> Good luck.
>
> --
> Mark Hadfield "Ka puwaha et tai nei, Hoea tatou"
> m.hadfield@niwa.co.nz
> National Institute for Water and Atmospheric Research (NIWA)
>
>
>
>

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying
