
Subject: Re: 8 to 24 bit conversion

Posted by [Robert Barnett](#) on Tue, 09 Nov 2004 23:36:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Firstly,

If there was anything I could wish for, it would be a 24bit buffer. I've had so much trouble trying to capture color snapshots of IDL windows (to send to a medical imaging PACS). This is particularly when I've rendered text and lines whose colors are not in the LUT. If anyone has a nice way around this, which is backwards compatible to IDL 5.1 I'd be so happy.

Currently I do as you do. I capture the screen and put it into my own byte array. I currently capture rendered text and lines with their own LUT separate from the image. Then I render the medical image in its LUT and capture that as well. I actually use 4 channels so as to include an alpha channel. I merge the two images together and then ship them off to wherever they are meant to go. It's very complicated and slow but it works and looks half reasonable. Retrospectively I would have split my colortable into two and scaled the medical image so it only uses its part of the color table.

Anyway, enough whining. Ken, is this what you wanted to do?

```
; Load the color table (blue)
loadct, 1
; Draw something at x=0, y=0
tvsc1, dist(100)
; Read the entire screen
image = tvrd()
; Read the look up table
tvlct, red, green, blue, /get
; Get the size of the snapshot of the screen
size = size(image)
; Assign a rgb image
rgb = bytarr(3,size[1],size[2])
; Fill in each channel based on the byte value of the image
rgb[0,*,*] = red[image]
rgb[1,*,*] = blue[image]
rgb[2,*,*] = green[image]
; Load a BW color table
loadct, 0
; Display each channel
tvsc1, rgb[0,0:100,0:100], 0 ; Red
tvsc1, rgb[1,0:100,0:100], 1 ; Green
tvsc1, rgb[2,0:100,0:100], 2 ; Blue
```

Ken Mankoff wrote:

```
>
> I am trying to convert an 8 bit image created in the Z buffer to a 24
> bit image. Is this possible? I would think so. But I am having trouble
> getting it to work.
>
> I based my code off of this algorithm:
> http://groups.google.com/groups?hl=en&lr=&selm=c0jq0j%24i40%241%40nntp6.u.washington.edu
>
> But I don't want to use the COLOR_QUAN bit of code that reduces it back
> to 8 bits.
>
> Can anyone point me to a library that does this, or point out some of
> the errors in the code below? Please don't point out the embarrassing
> optimization errors I know are there...
>
> Thanks
>
> -k.
>
>
> image = TVRD()
> TVLCT, R, G, B, /GET
>
> s = SIZE(image, /DIMENSIONS)
> rImage = BYTARR(s)
> gImage = BYTARR(s)
> bImage = BYTARR(s)
> ; replace this section with a HISTOGRAM statement
> for n=0, 255 do begin
>   idx = WHERE(image eq n, count)
>   if (count gt 0) then begin
>     rImage[idx] = R[n]
>     gImage[idx] = G[n]
>     bImage[idx] = B[n]
>   endif
> endfor
>
> newImage = bytarr( s(0), s(1), 3 )
> newImage[*,0] = rImage
> newImage[*,1] = gImage
```

```
> newImage[*,*,2] = bImage  
>
```

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223
