
Subject: Re: using TVRD(true=0) with a 24-bit image and decomposed=0
Posted by [Karl Schultz](#) on Mon, 29 Mar 2004 21:56:53 GMT

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

"Chris ODell" <odell@aos.wisc.edu> wrote in message
news:2fafbdc3.0403291218.95f4491@posting.google.com...
> To all IDL color-masters,
>
> I am trying to simply read an image that was originally plotted like
> this:
>
> tv, image
>
> where image is a 2D byte-array of colors IN MY CURRENT COLORTABLE; ie,
> numbers from 0 to 255. However, I am on a 24-bit display. I have
> device, decomposed=0
> and backing-store set on retain=2.
>
> If i do this:
>
> im2 = tvrd()
> tv, im2
>
> I get way more color=255 than I should; i do not get back the precise
> original image. I understand that I can get the RGB triples by doing
> tvrd(true=1), but *I don't want the RGB triples*. I just want the
> simple numbers from 0 to 255 that I originally plotted. How do I get
> them using tvrd()?

The IDL docs for TVRD say:

"If the display is a 24-bit display, and both the CHANNEL and TRUE
parameters are absent, the maximum RGB value in each pixel is returned."

This probably explains why you don't get back exactly what you put in and
why you see more 255 values.

If you do a TV with 8-bit image data and a color table on a 24-bit display,
IDL is going to translate your 8-bit data to 24 bits via the color table and
write the 24-bit data to the screen. There is not much else IDL can do
here. It needs to set all 24 bits to get the right picture on the screen.

Once the image is stored in the 24-bit frame buffer, the 8-bit data is lost,
from the point of view of the TV routine. Your IDL program could save it
someplace if it is needed later.

When IDL does the TVRD from the 24-bit frame buffer in this situation, it
needs to translate it back to 8 bits somehow. Clearly, there are some

difficulties here. One approach is to take the current color table and try to match every 24-bit pixel from the display to one of the color table entries. This would be extremely time consuming and may not always "work". The closest match may not be good enough, or if the color table contains two entries with the same color, which of the two should be used? So, IDL simply behaves as stated above.

Now, if you really wanted to, you could read all three channels with the TRUE kwd, and then use COLOR_QUAN to get a color array and palette back out of your 24-bit image. This is essentially the approach I mentioned above, which you can go ahead and do. But I think you'd be better off saving the original 8-bit data somehow.

Karl

Subject: Re: using TVRD(true=0) with a 24-bit image and decomposed=0
Posted by [Rick Towler](#) on Mon, 29 Mar 2004 22:51:06 GMT

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

"Chris ODell" wrote in message...

> To all IDL color-masters,
>
> I am trying to simply read an image that was originally plotted like
> this:
>
> tv, image
>
> where image is a 2D byte-array of colors IN MY CURRENT COLORTABLE; ie,
> numbers from 0 to 255. However, I am on a 24-bit display. I have
> device, decomposed=0
> and backing-store set on retain=2.
>
> If i do this:
>
> im2 = tvrd()
> tv, im2
>
> I get way more color=255 than I should; i do not get back the precise
> original image. I understand that I can get the RGB triples by doing
> tvrd(true=1), but *I don't want the RGB triples*. I just want the
> simple numbers from 0 to 255 that I originally plotted. How do I get
> them using tvrd()?

Read Karl's message, then download David Fanning's TVREAD program.

<http://www.dfanning.com/documents/programs.html#TVREAD>

-Rick

Subject: Re: using TVRD(true=0) with a 24-bit image and decomposed=0

Posted by [odell](#) on Wed, 31 Mar 2004 17:41:45 GMT

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

Thanks Karl...that makes a lot of sense about how IDL works. BTW, I wrote a simple replacement for TV that does what I want; it is a bit slower but it hasn't failed yet:

```
function TV_CT, Xo, Yo, Nx, Ny, _extra=_extra

if n_elements(Xo) eq 0 then Xo = 0
if n_elements(Yo) eq 0 then Yo = 0
if n_elements(Nx) eq 0 then Nx = !d.x_size - Xo
if n_elements(Ny) eq 0 then Ny = !d.y_size - Yo
im = tvrd(Xo, Yo, Nx, Ny, _extra = _extra, true=1)
imc = reform(im[0,*,*] + im[1,*,*]*256L + im[2,*,*]*256L^2)
tvlct, r, g, b, /get
ct = r + g*256L + b*256L^2
out = byte(imc)

for c = 0, 255 do begin
  w = where(imc eq ct[c])
  if w[0] ne -1 then out[w] = c
endfor

return, out
END
```

Chris

"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message news:<106h6sil1087e38@corp.supernews.com>...

> "Chris ODell" <odell@aos.wisc.edu> wrote in message

> news:2fafbdc3.0403291218.95f4491@posting.google.com...

>> To all IDL color-masters,

>>

>> I am trying to simply read an image that was originally plotted like

>> this:

>>

>> tv, image

>>

>> where image is a 2D byte-array of colors IN MY CURRENT COLORTABLE; ie,

>> numbers from 0 to 255. However, I am on a 24-bit display. I have

>> device, decomposed=0

>> and backing-store set on retain=2.

```
>>
>> If i do this:
>>
>> im2 = tvrd()
>> tv, im2
>>
>> I get way more color=255 than I should; i do not get back the precise
>> original image. I understand that I can get the RGB triples by doing
>> tvrd(true=1), but *I don't want the RGB triples*. I just want the
>> simple numbers from 0 to 255 that I originally plotted. How do I get
>> them using tvrd()?
>
> The IDL docs for TVRD say:
>
> "If the display is a 24-bit display, and both the CHANNEL and TRUE
> parameters are absent, the maximum RGB value in each pixel is returned."
>
> This probably explains why you don't get back exactly what you put in and
> why you see more 255 values.
>
> If you do a TV with 8-bit image data and a color table on a 24-bit display,
> IDL is going to translate your 8-bit data to 24 bits via the color table and
> write the 24-bit data to the screen. There is not much else IDL can do
> here. It needs to set all 24 bits to get the right picture on the screen.
>
> Once the image is stored in the 24-bit frame buffer, the 8-bit data is lost,
> from the point of view of the TV routine. Your IDL program could save it
> someplace if it is needed later.
>
> When IDL does the TVRD from the 24-bit frame buffer in this situation, it
> needs to translate it back to 8 bits somehow. Clearly, there are some
> difficulties here. One approach is to take the current color table and try
> to match every 24-bit pixel from the display to one of the color table
> entries. This would be extremely time consuming and may not always "work".
> The closest match may not be good enough, or if the color table contains two
> entries with the same color, which of the two should be used? So, IDL
> simply behaves as stated above.
>
> Now, if you really wanted to, you could read all three channels with the
> TRUE kwd, and then use COLOR_QUAN to get a color array and palette back out
> of your 24-bit image. This is essentially the approach I mentioned above,
> which you can go ahead and do. But I think you'd be better off saving the
> original 8-bit data somehow.
>
> Karl
```
