Subject: Re: findng array[3] in array[3,n]
Posted by John-David T. Smith on Thu, 25 Jan 2001 17:24:38 GMT
View Forum Message <> Reply to Message

tbowers wrote:
>
> If I have
>
> a=[$
> [0,1,2], $
> [3,4,5], $
> [6,7,8]]
>
> b=[3,4,5]
>
> how do I find where in a is the row vector b? The answer should
> be 1, the 2nd row of a. I've tried many
> permutations of where(), but I just don't get it. The only way I
> can get an answer is to loop through the rows till i find a match.
> What I'm really tryin' to do is to find a color in a color table,
> e.g. load RAINBOW color table, (loadct,13) then identify
> where a color is. For the color that's listed 6th in the color table
> (an almost black) that'd be the 5th row index. Like:

> So, does anyone know see an elegant solution to this problem??

I can't say it's elegant, but:

 where(total(rebin(b,n_elements(b),(size(a,/DIMENSIONS))[1],/ SAMP) eq
a,1) eq n_elements(b))

The rule is, when comparing arrays, you must expand the smaller to the
size of the larger.

JD

Subject: Re: findng array[3] in array[3,n]
Posted by Craig Markwardt on Thu, 25 Jan 2001 17:30:05 GMT
View Forum Message <> Reply to Message

"tbowers" <tbowers@nrlssc.navy.mil> writes:

> If I have
>
> a=[$
> [0,1,2], $
> [3,4,5], $

>   [6,7,8]]
>
> b=[3,4,5]
>
> how do I find where in a is the row vector b? The answer should
> be 1, the 2nd row of a. I've tried many
> permutations of where(), but I just don't get it. The only way I
> can get an answer is to loop through the rows till i find a match.
> What I'm really tryin' to do is to find a color in a color table,
> e.g. load RAINBOW color table, (loadct,13) then identify
> where a color is. For the color that's listed 6th in the color table
> (an almost black) that'd be the 5th row index. Like:

...

>

Your problem is that WHERE only does a 1D search. So you need to somehow convert your triplets to a single number.

Solution 1:

1. Convert your 3 BYTE values to a single LONG value:

```
    colorwords = long(r) + ishft(long(g),8L) + ishft(long(b),16L)
```

  do the same for your target, and use WHERE to find the match

  This will be fast if you need a few matches, slow if you need a ton of matches.

2. Search on R value alone using WHERE, then use a FOR loop to scan the resulting matches.

```
   wh = where(r EQ rtarg, ct)
   if ct GT 0 then for i = 0, ct-1 do if ...
```

   Probably overkill.

3. Use a Euclidean distance to find the color table entry with the smallest distance from you target value [rtarg, gtarg, btarg]. You should convert R G and B to vectors of type LONG to prevent overflow:

```
   dist = (r-rtarg)^2 + (g-gtarg)^2 + (b-btarg)^2
   wh = where(dist EQ min(dist))
```

   This will be the most robust to small variations in the color table (ie, if an exact match doesn't exist).

Any other ideas?

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Re: findng array[3] in array[3,n]
Posted by davidf on Thu, 25 Jan 2001 17:31:48 GMT

View Forum Message <> Reply to Message

JD Smith (jdsmith@astro.cornell.edu) writes:

> I can't say it's elegant, but:
>
> where(total(rebin(b,n_elements(b),(size(a,/DIMENSIONS))[1],/ SAMP) eq
> a,1) eq n_elements(b))
>
> The rule is, when comparing arrays, you must expand the smaller to the
> size of the larger.

I'm curious, JD. Are you any good with crossword puzzles?
I'm looking for a seven letter word that .... :-)

Cheers,

David

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

Subject: Re: findng array[3] in array[3,n]
Posted by davidf on Thu, 25 Jan 2001 17:48:21 GMT

View Forum Message <> Reply to Message

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

> Any other ideas?

Uh, use PickColorName in combination with FSC_Color? :-)

Cheers,

David

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

Subject: Re: findng array[3] in array[3,n]
Posted by T Bowers on Thu, 25 Jan 2001 22:53:14 GMT
View Forum Message <> Reply to Message

```
>  3. Use a Euclidean distance to find the color table entry with the
>     smallest distance from you target value [rtarg, gtarg, btarg].  You
>     should convert R G and B to vectors of type LONG to prevent
>     overflow:
>
>     dist = (r-rtarg)^2 + (g-gtarg)^2 + (b-btarg)^2
>     wh = where(dist EQ min(dist))
>
>     This will be the most robust to small variations in the color table
>     (ie, if an exact match doesn't exist).
```

Good Lord man! You're a genius!!!!

I'm learnig alot here. Your solution worked brilliantly for what I was
trying
to do. So... I went further and tried this.
Suppose I have an image that I got with tvrd(/true). This means it's a
[3,m,n]
image with the 1st dimension specifying r,g,b. Now, I wanna map this to an
IDLgrImage with transparency, so I wanna make it a [4,m,n] so' I can have me
and alpha channel. Say I want all black ([0,0,0]) to have an opacity of 0.
So I...

```
;say mapImage is [3,m,n] with lots of black background
sz = size(mapImage)
rgbaImage=bytarr(sz[1]+1,sz[2],sz[3]) ; make it [4,m,n]; r,g,b,alpha
rgbaImage[0:2,*,*] = mapImage

;start w/ all opaque (255)
rgbaImage[3,*,*] = 255
```

```
; this color will be transparent
xparentcolor = [000,000,000]

;get rgb channels, which are 2D
r = long(rgbaImage[0,*,*]) & g = long(rgbaImage[1,*,*]) & b =
long(rgbaImage[2,*,*])

;do Craig's magic
eucDist = (r-xparentcolor[0])^2 + (g-xparentcolor[1])^2 +
(b-xparentcolor[2])^2
colorIndex = where(eucDist EQ min(eucDist), count)

;set the alpha channel indices = 0 where theres black
rgbaImage[3,colorIndex] = 0

;now it's readt for my IDLgrImage
sState.oMapImage->setProperty, DATA=rgbaImage, HIDE=0
```

Problem is is that my black is still opaque. I still can't see my objects behind it. Is it my misunderstanding of bringing this to 3D that's illing me?
Looks like it should work.

Thanks to all who responded.
Oh, and yes David, I was gonna use your code if I couldn't find a way to save a function call. Thanks.

Subject: Re: findng array[3] in array[3,n]
Posted by Craig Markwardt on Fri, 26 Jan 2001 13:42:20 GMT
View Forum Message <> Reply to Message

"tbowers" <tbowers@nrlssc.navy.mil> writes:

```
>
... long description deleted ...
>  colorIndex = where(eucDist EQ min(eucDist), count)
>
>  ;set the alpha channel indices = 0 where theres black
>  rgbaImage[3,colorIndex] = 0
>
>  ;now it's readt for my IDLgrImage
>  sState.oMapImage->setProperty, DATA=rgbaImage, HIDE=0
>
>  Problem is is that my black is still opaque. I still can't see my objects
>  behind it. Is it my misunderstanding of bringing this to 3D that's illing
>  me?
```

Before you blame oMapImage, are you sure you actually got the alpha channel you were hoping for?

One cool thing about the Euclidean distance thing is that you don't have to take just the minimum distance. If you know that "black" can be several nearby colors then you can say

```
 colorIndex = where((eucDist-min(eucDist)) LT tol^2, count)
```

where tol is the number of deviant color levels you are willing to accept. This helps especially in photographic pictures where solid swatches are rarely a single color value.

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------