Subject: Re: Array Subscripting Puzzle
Posted by James Kuyper on Fri, 17 May 2002 18:15:37 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Folks,
>
> I have a 24-bit image. You can interleave it anyway
> you like that will make the problem described below
> trackable. At the moment it is 800 by 600 by 3.
>
> I have the indices of something I want to draw on
> the image. Say they are the indices of the outlines
> of some continents. For example, like this:
>
> window, xsize=800, ysize=600
> map_set, /Cylindrical, position=[0,0,1,1]
> map_continents, /fill
> a = tvrd()
> indices = where(a GT 0)
>
> I want to make all the outline pixels yellow.
> I *could* do this:
>

> r = Reform((image[*,*,0]))
> g = Reform((image[*,*,1]))
> b = Reform((image[*,*,2]))
> r[indices] = 255
> g[indices] = 255
> b[indices] = 0
> image[*,*,0] = r
> image[*,*,1] = g
> image[*,*,2] = b
>
> That seems wasteful and inelegant. There must be
> a way to do this in one go. I'm sure it uses REBIN
> and REFORM, but I'm not sure in which order. :-(
>
> Can anyone help?

How about this:

m = a GT 0
image = image*(1-m)+[[[255*m]],[[255*m]],[[0*m]]]

Subject: Re: Array Subscripting Puzzle
Posted by Liam E. Gumley on Fri, 17 May 2002 18:30:06 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Folks,
>
> I have a 24-bit image. You can interleave it anyway
> you like that will make the problem described below
> trackable. At the moment it is 800 by 600 by 3.
>
> I have the indices of something I want to draw on
> the image. Say they are the indices of the outlines
> of some continents. For example, like this:
>
> window, xsize=800, ysize=600
> map_set, /Cylindrical, position=[0,0,1,1]
> map_continents, /fill
> a = tvrd()
> indices = where(a GT 0)
>
> I want to make all the outline pixels yellow.
> I *could* do this:
>
> r = Reform((image[*,*,0]))
> g = Reform((image[*,*,1]))
> b = Reform((image[*,*,2]))
> r[indices] = 255
> g[indices] = 255
> b[indices] = 0
> image[*,*,0] = r
> image[*,*,1] = g
> image[*,*,2] = b
>
> That seems wasteful and inelegant. There must be
> a way to do this in one go. I'm sure it uses REBIN
> and REFORM, but I'm not sure in which order. :-(
>
> Can anyone help?

Here's a shorcut with no array rearrangment.
Note that the 24-bit image must be in [NCOL, NROW, 3] format:

```
dims = size(image, /dimensions)
ncol = dims[0]
nrow = dims[1]
chan = 0 ; red channel
image[indices + (chan * ncol * nrow)] = 255
chan = 1 ; green channel
```

image[indices + (chan * ncol * nrow)] = 255
chan = 2 ; blue channel
image[indices + (chan * ncol * nrow)] = 0

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Array Subscripting Puzzle
Posted by dmarshall on Fri, 17 May 2002 18:57:58 GMT
View Forum Message <> Reply to Message

This wouldn't work anyways will it since indices references a which is a
large (800x600x3) linear array whereas r,g,b are only (800x600). (?)

You could "collapse" image down to a pseudo b/w
 bwimage=image[*,*,0]+image[*,*,1]+image[*,*,2]
 bwImage=reform(bwImage, /overwrite) ;make sure bwimage is 2D

Reform image so it is same as bwimage
 image=reform(image,800*600,3, /overwrite)
Yellowize
 image[where(bwimage GT 0),*]=[255,255,0]
Reform back
 image=reform(image,800,600,3, /overwrite)

bwimage and the operations must be forced to long since you will get values
greater than 255.

Dave.
> I have the indices of something I want to draw on
> the image. Say they are the indices of the outlines
> of some continents. For example, like this:
>
>   window, xsize=800, ysize=600
>   map_set, /Cylindrical, position=[0,0,1,1]
>   map_continents, /fill
>   a = tvrd()
>   indices = where(a GT 0)
>
> I want to make all the outline pixels yellow.
> I *could* do this:
>
>   r = Reform((image[*,*,0]))
>   g = Reform((image[*,*,1]))
>   b = Reform((image[*,*,2]))

>   r[indices] = 255
>   g[indices] = 255
>   b[indices] = 0
>   image[*,*,0] = r
>   image[*,*,1] = g
>   image[*,*,2] = b
>
> That seems wasteful and inelegant. There must be
> a way to do this in one go. I'm sure it uses REBIN
> and REFORM, but I'm not sure in which order. :-(
>
> Can anyone help?
>
> 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: Array Subscripting Puzzle
Posted by James Kuyper on Fri, 17 May 2002 19:45:10 GMT
View Forum Message <> Reply to Message

dmarshall@ivory.trentu.ca wrote:

> This wouldn't work anyways will it since indices references a which is a
> large (800x600x3) linear array whereas r,g,b are only (800x600). (?)
...

>> I have the indices of something I want to draw on
>> the image. Say they are the indices of the outlines
>> of some continents. For example, like this:
>>
>>   window, xsize=800, ysize=600
>>   map_set, /Cylindrical, position=[0,0,1,1]
>>   map_continents, /fill
>>   a = tvrd()

'a' was created by calling tvrd(), not tvrd(/TRUE). Therefore, it will
be (800x600).

Subject: Re: Array Subscripting Puzzle
Posted by Richard Younger on Fri, 17 May 2002 21:18:15 GMT
View Forum Message <> Reply to Message

I came up with an answer for a [3,800,600] image, but couldn't quite
wrap my head around the [800,600,3], so I swapped:

```
 image = TRANSPOSE(image, [2,0,1])

 color_vec = [255,255,0]
 mask = [[3*indices], [(3*indices + 1)],[(3*indices + 2)]]
 image[mask] = $
  REBIN(TRANSPOSE(color_vec), N_ELEMENTS(indices), 3)
```

It certainly would be uglier if you made it all one line (well, two with
TRANSPOSE). Someone clever could probably swap around the index order
on the image to eliminate that transpose and pretty up the mask
construction, too.

Good Luck,
Rich

--
Richard Younger

David Fanning wrote:
>
> Folks,
>
> I have a 24-bit image. You can interleave it anyway
> you like that will make the problem described below
> trackable. At the moment it is 800 by 600 by 3.

[...]

> That seems wasteful and inelegant. There must be
> a way to do this in one go. I'm sure it uses REBIN
> and REFORM, but I'm not sure in which order. :-(
>
> Can anyone help?
>
> Cheers,
>
> David
>

Subject: Re: Array Subscripting Puzzle
Posted by JD Smith on Wed, 22 May 2002 21:12:39 GMT
View Forum Message <> Reply to Message

On Fri, 17 May 2002 14:18:15 -0700, Richard Younger wrote:

> I came up with an answer for a [3,800,600] image, but couldn't quite
> wrap my head around the [800,600,3], so I swapped:
>
> image = TRANSPOSE(image, [2,0,1])
>
> color_vec = [255,255,0]
> mask = [[3*indices], [(3*indices + 1)],[(3*indices + 2)]] image[mask] =
> $
>   REBIN(TRANSPOSE(color_vec), N_ELEMENTS(indices), 3)
>
> It certainly would be uglier if you made it all one line (well, two with
> TRANSPOSE). Someone clever could probably swap around the index order
> on the image to eliminate that transpose and pretty up the mask
> construction, too.

Ahh yes, you point out a simplification in my index computation for
3x800x600.

3*(y*s[0]+x)==3*inds

So that I could have written:

```
image[rebin(1#(3*inds),3,n)+rebin(indgen(3),3,n)]= $
  rebin([255,255,0],3,n)
```

Notice that Richard's example proves the point that you can use any
convenient intermediary format: he chose nx3, in contrast to 3xn for this
problem.

JD