
Subject: Re: Maximum Intensity without FOR Loop
Posted by [David Fanning](#) on Tue, 11 May 2004 19:29:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stefan Tuchschnid writes:

> Hi there
> a lot of posts talk about the necessity to avoid FOR loops ('if you
> use for loops, make sure a lot of stuff is happening in there...').
> However, I can't seem to find a better & working solution for the
> following code fragment:
>
> FOR i=0,x_resolution-1 DO BEGIN
> FOR j=0,y_resolution-1 DO BEGIN
> reformat_data[i,j,k]=MAX(data[i,j,lower_limit:upper_limit])
> ENDFOR
> ENDFOR
>
> Background: data is a image stack (MRI 3D Data), we would like to find
> the MIP (Maximum Intesity Projection) over a certain number of images.
> The current solution is too slow, and a better solution highly
> appreciated!
> Algorithm-Freaks? Anyone?

I should think something like this would work:

```
mip = Max(data[*,*],lower_limit:upper_limit], Dimension=3)
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Maximum Intensity without FOR Loop
Posted by [btt](#) on Tue, 11 May 2004 19:46:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stefan Tuchschnid wrote:

> Hi there
> a lot of posts talk about the necessity to avoid FOR loops ('if you
> use for loops, make sure a lot of stuff is happening in there...').
> However, I can't seem to find a better & working solution for the
> following code fragment:
>

```
> FOR i=0,x_resolution-1 DO BEGIN
>   FOR j=0,y_resolution-1 DO BEGIN
>     reformat_data[i,j,k]=MAX(data[i,j,lower_limit:upper_limit])
>   ENDFOR
> ENDFOR
>
> Background: data is a image stack (MRI 3D Data), we would like to find
> the MIP (Maximum Intesity Projection) over a certain number of images.
```

Hi,

```
IDL> data = Bindgen(3,4,5)
```

You could use the DIMENSION keyword to MAX to find the maximum in that dimension (across the entire array.)

```
IDL> print, max(data, dim = 3)
 48 49 50
 51 52 53
 54 55 56
 57 58 59
```

Or, you could find the maximum using the 'max' operator on specified dimension (you can do more than two at a time.)

```
IDL> print, data[:,*,0] > data[:,*,2]
 24 25 26
 27 28 29
 30 31 32
 33 34 35
```

Ben
