
Subject: Re: Using MIN on arrays : Exorcising loops?
Posted by [Craig Markwardt](#) on Fri, 05 Oct 2001 13:03:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Andrew Cool <cooladjc@chariot.net.au> writes:

```
> Hi all,  
>  
> I have three slabs of data [640,500] held as an array [640,500,3].  
>  
> I need to populate another 2D array [640,500] with the minimum  
> value for every x,y coordinate found in the first 3 arrays.  
>  
> At the moment my code loops something like this :-  
>  
> data_array = Fltarr(640,500,3)  
> Min_array = Fltarr(640,500)  
>  
> For x = 0,639 Do Begin  
>   For y = 0,249 Do Begin  
>     Min_array(x,y) = MIN(data_array(x,y,*))  
>   End  
> End  
>  
> There's gotta be a better way, surely? Some syntax variant on MIN?  
> Histogram even? ;-)
```

Hi Andrew--

This is a pretty good question, and "no," I don't think there is a way to do it with the MIN() function alone. I have a routine called CMAPPLY on my web page which can do this kind of thing, but it won't be terribly efficient here because it still does a loop based on the number of output elements, so it is exactly the same as the loop you posted above.

Still there is a trivial way to solve this using the threshold operator. I have always advocated that loops are not "bad," rather the poor use of loops is bad. :-) The trick here is to put the most expensive operation -- operating on an image worth of data -- at the center of your loop.

How does this work for you?

```
data_array = Fltarr(640,500,NZ)  
Min_array = data_array(*,*,0)  
  
for i = 1, NZ-1 do $
```

```
min_array = min_array < data_array(*,*,i)
```

Since NZ is 3, this really only has three iterations, and most of the work is done by the "<" operator, which is the same as MIN().

Hmm, maybe I should put something like this in CMAPPLY...

Craig

--

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