Subject: Re: Maximum value array resampling
Posted by Richard French on Fri, 05 Aug 2005 20:17:19 GMT
View Forum Message <> Reply to Message

Well, the algorithm I supplied works but is not very fast. Instead of rebinning up, it would be faster to subsample down and make the comparison between images of the desired final size. I should have thought of that before. I'll see if it is faster than the loop method in the end. For a test case that I just ran, the loop method is faster than the rebinning method.
Sorry! I'll get back to you with revised code and time tests.
Dick

```
>
>> I'm trying to figure out an IDL-efficient way to resample a series of
>> images. I know how to do this is a ruinously laborious fashion using
>> loops but I know there's any easier way.
>>
>> Consider the following 4x4 array:
>>
>> x = [$
>> [0,3,4,5],$
>> [1,2,7,0],$
>> [3,2,9,0],$
>> [7,0,5,6]]
>>
>> I want to resample this to y, a 2x2 array. Each element would contain
>> the maximum value of the corresponding 4 pixels. y would then look like
>>
>> [$
>> [3,7],$
>> [7,9]]
>>
>> So element [0,0] in y is max(x[0:1,0:1]) etc. As I understand it,
>> rebin/congrid won't do this. Each image is about 5000x2000 and there
>> are several hundred to process.
>>
>> Thanks!
>>
>
> Try this (I don't know how it scales with image size, but give it a try!)
>
> The basic idea is to make four rebinned copies of the image  with the upper
> left, upper right, lower left, and lower right pixels in each 2x2 pair and
> to replace the array elements by the largest value as you compare each of
> the rebinned images in turn. Probably you can cut down on memory use by
> using TEMPORARY() and there may be other increases in efficiency as well.
> Good luck!
>
```

```
> Dick French
>
>
> x=[$
> [0,3,4,5,3,7],$
> [1,2,7,0,2,9],$
> [3,2,9,0,4,2],$
> [7,0,5,6,1,5]]
>
> Print,x
>
> ; get size
>
> size=size(x,/DIM)
>
> nx=size[0]
> ny=size[1]
>
> nx2=nx/2
> ny2=ny/2
>
> ; get indices of upper left element of each 2x2 cell
>
>  l=rebin(nx*2#lindgen(ny2),nx2,ny2)+rebin(lindgen(nx2)#2,nx2, ny2)
>
>
> ; get indices of ul,ur, ll, lr of each 2x2 cell
> offsets=[0,1,nx,nx+1]
>
> for n=0,3 do begin
>         xtemp=rebin(x[l+offsets[n]],nx,ny,/sample)
>         x=x+((xtemp-x)>0)   ; replace x by current largest value
> endfor
>
> xfinal=rebin(x,nx2,ny2)
> print
> print,xfinal
>
> End
>
> IDL> .run trythis
>        0       3       4       5       3       7
>        1       2       7       0       2       9
>        3       2       9       0       4       2
>        7       0       5       6       1       5
>
>        3       7       9
>        7       9       5
```

>
>