Subject: Re: Finding the mean of a set of images
Posted by Jaco van Gorkom on Tue, 22 Oct 2002 16:33:36 GMT

View Forum Message <> Reply to Message

"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message
news:onznt6k38x.fsf@cow.physics.wisc.edu...
```
>  David Oesch <oesch@giub.unibe.ch> writes:
>>   ...
>>  Does anyone have an algorithm for finding the mean/standardeviation etc
>>  at each pixel position for a set of equal size 2-D images? Currently the
>>  only way I have to do this is to extract all the values for a given
>>  pixel position into a 1-D array and find the mean/standardeviation etc
>>  on that. Doing it pixel by pixel like this is inefficient in IDL so I am
>>  looking for an *array* based algorithm that would find all
>>  the mean/standardeviation etc in parallel.    ...
>
>  Sure, if you stack your image into a 3D image cube, then you would
>  have something like IMAGE = FLTARR(NX, NY, NIMAGES)
>
>  Then the mean image is:
>
>    mean = total(image,3)/nimages
>
>  The standard deviation is:
>
>    meancube = rebin(reform(mean,nx,ny,1),nx,ny,nimages)
>    std = sqrt(total((image - meancube)^2,3)/(nimages-1))
>
>  Now, what you meant by "etc" can get a little hairier.  If you want to
>  do median you are probably in trouble, but min and max are easy too:
>
>    minimage = image(*,*,0)
>    maximage = minimage
>    for i = 1, nimages-1 do begin
>      minimage = minimage < image(*,*,i)
>      maximage = maximage > image(*,*,i)
>    endfor
```

I believe that IDL 5.5 offers the luxury of
```
  maximage = MAX(image, MIN=minimage, DIMENSION=3)
```

As for median, well, there was a thread on "Finding the median of a set of images" back in '96. Should be valid still, I guess. Improvising with transpose(), reform(), median(image, Nimages) and rebin() should be fun, but not be very fast.

Jaco