

---

Subject: Re: Finding the mean of a set of images  
Posted by [David Oesch](#) on Tue, 22 Oct 2002 15:23:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Craig Markwardt wrote:

```
> David Oesch <oesch@giub.unibe.ch> writes:
>
>> Hello outthere,
>>
>> I know this topic was up before, but all I could find in the list was to
>> go for CALL_EXTERNAL and use a FORTRAN etc. program. Here's my problem:
>> Does anyone have an algorithm for finding the mean/standarddeviation 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/standarddeviation 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/standarddeviation etc in parallel.
>> Any progs so far in IDL for this problem?..or a decent fortran or C program?
>>
>
>
> 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
>
> It's a loop, but unless you have a bazillion images, it will be fast.
>
> Craig
```

>

Yes, I got that one with TOTAL, the problem is, it only works with imagesets where all datalayers consists of 100% valid pixels...but this is not always the case...  
to eliminate those values, I came as far as

```
mean = total((image NE -9999),3)/nimages
```

but the problem is, nimages needs to be changed to...

up to now, I use the following simple function to get to my data...but it is time consuming ..

```
function mean3d,file,nvalid=nvalid
```

```

,*****
;
; finding the mean at each pixel position for a set of equal size 2-D images
,*****
;
;file= 3d array
;nvalid= novalid values
;calling sequence: a=mean3d(array,nvalid=-9999)
```

```
s=SIZE(file)
outarray=FLTARR(s(1),s(2))
```

```
i=0!
FOR i=0,fix(s(1))-1 DO BEGIN
  j=0!
  FOR j=0,fix(s(2))-1 DO BEGIN
    check=WHERE(file(i,j,*) NE nvalid,count_check)
    IF count_check GT 0 THEN BEGIN
      outarray(i,j)=MEAN(file(i,j,check))
    ENDIF
  
```

```
  ENDFOR
ENDFOR
RETURN,outarray
```

```
END
```