Subject: Re: Calculate the mean of many images
Posted by greg.addr on Sun, 16 Nov 2008 08:08:21 GMT
View Forum Message <> Reply to Message

How about:

q=where(QA eq 1, n)
mean=total(image[q])/n

cheers,
Greg

Subject: Re: Calculate the mean of many images
Posted by Craig Markwardt on Sun, 16 Nov 2008 08:54:32 GMT
View Forum Message <> Reply to Message

On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
> Hello,
> I know this topic has been posted several times. But I could not find
> my answer from these posted.
> My issue is: I have many images 2 bands in each, one image is QA image
> and the other one is data.
> I need to calculate the mean of good pixels. Let's say QA image tells
> me the location of good pixels, e.g. 1 for good pixels, and other for
> bad. There are also NaN values. So, if the pixels are "good" in 7
> images out of ten, then
> (pixel1+pixel2...+Pixel7) /7

I would loop over input images, and keep track of the cumulative sum
of the number of valid pixels (NPIX), and the cumulative sum of the
pixel values (SUM). Something like the following. Since there are
only a few images, there will be very little overhead in the FOR-loop.

Craig

```
npix = 0 & sum = 0
for i = 0, n_images-1 do begin
  qa = ... the ith QA image ...
  img = ... the ith image ...
  mask = (qa EQ 1) AND (finite(img) EQ 1)

  ;; Sum valid pixels
  npix += mask
  wh = where(mask, ct)
  if ct GT 0 then sum(wh) += img(wh)
endfor
```

```
;; Positions where there are good pixels
qa_avg = (npix GT 0)
wh = where(qa_avg EQ 1)

;; Compute average for valid pixels
avg = sum*0
avg(wh) = sum(wh) / npix(wh)
```

Subject: Re: Calculate the mean of many images
Posted by Jeremy Bailin on Sun, 16 Nov 2008 14:57:55 GMT
View Forum Message <> Reply to Message

On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
> Hello,
> I know this topic has been posted several times. But I could not find
> my answer from these posted.
> My issue is: I have many images 2 bands in each, one image is QA image
> and the other one is data.
> I need to calculate the mean of good pixels. Let's say QA image tells
> me the location of good pixels, e.g. 1 for good pixels, and other for
> bad. There are also NaN values. So, if the pixels are "good" in 7
> images out of ten, then
> (pixel1+pixel2...+Pixel7) /7
>
> How can I do that?
>
> Thanks any advice!
>
> Elkunn

I'm going to assume that qa and image are both nx x ny x nimage
arrays, and qa is 1 for good, 0 for bad.

```
ngoodpix = total(qa, 3)
imagesum = total(image * qa, 3)
meanimage = imagesum/ngoodpix
```

-Jeremy.

Subject: Re: Calculate the mean of many images
Posted by Wasit.Weather on Sun, 16 Nov 2008 18:30:20 GMT
View Forum Message <> Reply to Message

On Nov 16, 2:54 am, Craig Markwardt <cbmarkwa...@gmail.com> wrote:

> On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>> Hello,
>> I know this topic has been posted several times. But I could not find
>> my answer from these posted.
>> My issue is: I have many images 2 bands in each, one image is QA image
>> and the other one is data.
>> I need to calculate the mean of good pixels. Let's say QA image tells
>> me the location of good pixels, e.g. 1 for good pixels, and other for
>> bad. There are also NaN values. So, if the pixels are "good" in 7
>> images out of ten, then
>> (pixel1+pixel2...+Pixel7) /7
>
> I would loop over input images, and keep track of the cumulative sum
> of the number of valid pixels (NPIX), and the cumulative sum of the
> pixel values (SUM). Something like the following. Since there are
> only a few images, there will be very little overhead in the FOR-loop.
>
> Craig
>
> npix = 0 & sum = 0
> for i = 0, n_images-1 do begin
> qa = ... the ith QA image ...
> img = ... the ith image ...
> mask = (qa EQ 1) AND (finite(img) EQ 1)
>
> ;; Sum valid pixels
> npix += mask
> wh = where(mask, ct)
> if ct GT 0 then sum(wh) += img(wh)
> endfor
>
> ;; Positions where there are good pixels
> qa_avg = (npix GT 0)
> wh = where(qa_avg EQ 1)
>
> ;; Compute average for valid pixels
> avg = sum*0
> avg(wh) = sum(wh) / npix(wh)

Thanks for all the comments. I think Craig's method would work for me.
the Actual QA image contains more than one value, such as 0.00 (good
pixel), 1.000 (between good and band), 2.000 (cloud), 3.000(snow),
etc.
What does finite(img) EQ 1 mean here? Can I write this statement as
the following?

mask = (qa EQ 1.000 and qa EQ 0.000) AND (finite(img) EQ 1.000 &&

0.000)

Thanks

Subject: Re: Calculate the mean of many images
Posted by Wasit.Weather on Sun, 16 Nov 2008 18:32:12 GMT
View Forum Message <> Reply to Message

On Nov 16, 12:30 pm, Bulrush <Wasit.Weat...@gmail.com> wrote:
> On Nov 16, 2:54 am, Craig Markwardt <cbmarkwa...@gmail.com> wrote:
>
>
>
>
>
>> On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>>> Hello,
>>> I know this topic has been posted several times. But I could not find
>>> my answer from these posted.
>>> My issue is: I have many images 2 bands in each, one image is QA image
>>> and the other one is data.
>>> I need to calculate the mean of good pixels. Let's say QA image tells
>>> me the location of good pixels, e.g. 1 for good pixels, and other for
>>> bad. There are also NaN values. So, if the pixels are "good" in 7
>>> images out of ten, then
>>> (pixel1+pixel2...+Pixel7) /7
>
>> I would loop over input images, and keep track of the cumulative sum
>> of the number of valid pixels (NPIX), and the cumulative sum of the
>> pixel values (SUM). Something like the following. Since there are
>> only a few images, there will be very little overhead in the FOR-loop.
>
>> Craig
>
>> npix = 0 & sum = 0
>> for i = 0, n_images-1 do begin
>> qa = ... the ith QA image ...
>> img = ... the ith image ...
>> mask = (qa EQ 1) AND (finite(img) EQ 1)
>
>> ;; Sum valid pixels
>> npix += mask
>> wh = where(mask, ct)
>> if ct GT 0 then sum(wh) += img(wh)
>> endfor
>

>> ;; Positions where there are good pixels
>> qa_avg = (npix GT 0)
>> wh = where(qa_avg EQ 1)
>
>> ;; Compute average for valid pixels
>> avg = sum*0
>> avg(wh) = sum(wh) / npix(wh)
>
> Thanks for all the comments. I think Craig's method would work for me.
> the Actual QA image contains more than one value, such as 0.00 (good
> pixel), 1.000 (between good and band), 2.000 (cloud), 3.000(snow),
> etc.
> What does finite(img) EQ 1 mean here? Can I write this statement as
> the following?
>
> mask = (qa EQ 1.000 and qa EQ 0.000) AND (finite(img) EQ 1.000 &&
> 0.000)
>
> Thanks- Hide quoted text -
>
> - Show quoted text -

Craig, is there Where routine is missing in the statement?

thanks

Subject: Re: Calculate the mean of many images
Posted by Chris[6] on Sun, 16 Nov 2008 19:08:39 GMT
View Forum Message <> Reply to Message

> What does finite(img) EQ 1 mean here? Can I write this statement as
> the following?

finitie(img) eq 1 will evaluate to the same thing as finite(img) - the
finite function returns a 1 or a zero depending on whether the input
is finite.

> mask = (qa EQ 1.000 and qa EQ 0.000) AND (finite(img) EQ 1.000 &&
> 0.000)

this statement is mutually exclusive (qa can't equal 1 and zero at the
same time). The finite part is equally problematic. I'm not sure what
you want mask to be. But anyways, once you know how to properly phrase
the logical test for your mask, I would do the following:

;- calculate the average
mask = ... logical test on QA which is TRUE for a pixel which you

want to consider ...
average_image = total(im * mask, 3) / total(mask, 3)

;- check for and pixels where there were no good images
bad = where(total(mask,3) eq 0, ct)
if ct eq 0 then average_image[bad] = !values.f_nan

maybe a good mask would look like
mask = finite(qa) if anything finite is good or,
if you want to pull out specific qa values,
mask = ((qa eq 1) or (qa eq 2))

chris

Subject: Re: Calculate the mean of many images
Posted by Craig Markwardt on Sun, 16 Nov 2008 20:04:07 GMT
View Forum Message <> Reply to Message

On Nov 16, 1:30 pm, Bulrush <Wasit.Weat...@gmail.com> wrote:
> On Nov 16, 2:54 am, Craig Markwardt <cbmarkwa...@gmail.com> wrote:
>
>
>
>> On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>>> Hello,
>>> I know this topic has been posted several times. But I could not find
>>> my answer from these posted.
>>> My issue is: I have many images 2 bands in each, one image is QA image
>>> and the other one is data.
>>> I need to calculate the mean of good pixels. Let's say QA image tells
>>> me the location of good pixels, e.g. 1 for good pixels, and other for
>>> bad. There are also NaN values. So, if the pixels are "good" in 7
>>> images out of ten, then
>>> (pixel1+pixel2...+Pixel7) /7
>
>> I would loop over input images, and keep track of the cumulative sum
>> of the number of valid pixels (NPIX), and the cumulative sum of the
>> pixel values (SUM). Something like the following. Since there are
>> only a few images, there will be very little overhead in the FOR-loop.
>
>> Craig
>
>> npix = 0 & sum = 0
>> for i = 0, n_images-1 do begin
>> qa = ... the ith QA image ...
>> img = ... the ith image ...

>> mask = (qa EQ 1) AND (finite(img) EQ 1)
>
>> ;; Sum valid pixels
>> npix += mask
>> wh = where(mask, ct)
>> if ct GT 0 then sum(wh) += img(wh)
>> endfor
>
>> ;; Positions where there are good pixels
>> qa_avg = (npix GT 0)
>> wh = where(qa_avg EQ 1)
>
>> ;; Compute average for valid pixels
>> avg = sum*0
>> avg(wh) = sum(wh) / npix(wh)
>
> Thanks for all the comments. I think Craig's method would work for me.
> the Actual QA image contains more than one value, such as 0.00 (good
> pixel), 1.000 (between good and band), 2.000 (cloud), 3.000(snow),
> etc.
> What does finite(img) EQ 1 mean here? Can I write this statement as
> the following?
>
> mask = (qa EQ 1.000 and qa EQ 0.000) AND (finite(img) EQ 1.000 &&
> 0.000)

As Chris points out, this is a logically inconsistent expression.
Probably you want,
  (qa EQ 1 OR qa EQ 0)
for the first part of your expression.

I'm not sure what you are getting at from the second part of the
expression. FINITE(img) is 0 if the pixel value is infinite or NaN, 1
otherwise. Looks like your "&& 0.000" is superfluous.

To answer your other question: the WHERE occurs a few lines later.
I'm using a MASK since you need to accumulate the number of pixels as
well as the pixel values.

Subject: Re: Calculate the mean of many images
Posted by Wasit.Weather on Sun, 16 Nov 2008 20:52:48 GMT
View Forum Message <> Reply to Message

On Nov 16, 2:04 pm, Craig Markwardt <cbmarkwa...@gmail.com> wrote:
> On Nov 16, 1:30 pm, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>

>
>> On Nov 16, 2:54 am, Craig Markwardt <cbmarkwa...@gmail.com> wrote:
>
>>> On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>>>> Hello,
>>>> I know this topic has been posted several times. But I could not find
>>>> my answer from these posted.
>>>> My issue is: I have many images 2 bands in each, one image is QA image
>>>> and the other one is data.
>>>> I need to calculate the mean of good pixels. Let's say QA image tells
>>>> me the location of good pixels, e.g. 1 for good pixels, and other for
>>>> bad. There are also NaN values. So, if the pixels are "good" in 7
>>>> images out of ten, then
>>>> (pixel1+pixel2...+Pixel7) /7
>
>>> I would loop over input images, and keep track of the cumulative sum
>>> of the number of valid pixels (NPIX), and the cumulative sum of the
>>> pixel values (SUM). Something like the following. Since there are
>>> only a few images, there will be very little overhead in the FOR-loop.
>
>>> Craig
>
>>> npix = 0 & sum = 0
>>> for i = 0, n_images-1 do begin
>>> qa = ... the ith QA image ...
>>> img = ... the ith image ...
>>> mask = (qa EQ 1) AND (finite(img) EQ 1)
>
>>> ;; Sum valid pixels
>>> npix += mask
>>> wh = where(mask, ct)
>>> if ct GT 0 then sum(wh) += img(wh)
>>> endfor
>
>>> ;; Positions where there are good pixels
>>> qa_avg = (npix GT 0)
>>> wh = where(qa_avg EQ 1)
>
>>> ;; Compute average for valid pixels
>>> avg = sum*0
>>> avg(wh) = sum(wh) / npix(wh)
>
>> Thanks for all the comments. I think Craig's method would work for me.
>> the Actual QA image contains more than one value, such as 0.00 (good
>> pixel), 1.000 (between good and band), 2.000 (cloud), 3.000(snow),
>> etc.
>> What does finite(img) EQ 1 mean here? Can I write this statement as

>> the following?
>
>> mask = (qa EQ 1.000 and qa EQ 0.000) AND (finite(img) EQ 1.000 &&
>> 0.000)
>
> As Chris points out, this is a logically inconsistent expression.
> Probably you want,
> (qa EQ 1 OR qa EQ 0)
> for the first part of your expression.
>
> I'm not sure what you are getting at from the second part of the
> expression. FINITE(img) is 0 if the pixel value is infinite or NaN, 1
> otherwise. Looks like your "&& 0.000" is superfluous.
>
> To answer your other question: the WHERE occurs a few lines later.
> I'm using a MASK since you need to accumulate the number of pixels as
> well as the pixel values.

Thank you. I just run the code, but it always reporting syntax error
for sum(wh) += img(wh), for the brackets.

Subject: Re: Calculate the mean of many images
Posted by David Fanning on Sun, 16 Nov 2008 21:04:32 GMT
View Forum Message <> Reply to Message

Bulrush writes:

> Thank you. I just run the code, but it always reporting syntax error
> for sum(wh) +=3D img(wh), for the brackets.

I think that is a hold-over from Craig's days as the King
of IDL 4.0. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Calculate the mean of many images
Posted by Wasit.Weather on Mon, 17 Nov 2008 00:35:10 GMT

On Nov 16, 3:04 pm, David Fanning <n...@dfanning.com> wrote:
> Bulrush writes:
>> Thank you. I just run the code, but it always reporting syntax error
>> for sum(wh) +=3D img(wh), for the brackets.
>
> I think that is a hold-over from Craig's days as the King
> of IDL 4.0. ;-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Seems like I got lost between the codes. the mean I got with avg[wh]
is all zero. Here is my code. Thanks for your help.

```
FOR K = 0, images-1 DO BEGIN
  ENVI_OPEN_FILE, files[K], r_fid=files_fid
  ENVI_FILE_QUERY, files_fid, .....
  pos = lindgen(nb)
  ; get the data
  Data = ENVI_GET_DATA(fid=files_fid, dims=dims, pos=pos[0])
  QA = ENVI_GET_DATA(fid=files_fid, dims=dims, pos=pos[1])
  mask = (QA EQ 0.00) AND (finite(Data) EQ 1) OR (QA EQ 1.00) AND
(finite(Data) EQ 1)

; Sum Valid Pixels
  npix += mask
  wh = where(mask, ct)
  if ct GT 0 then sum[wh] += Data[wh]
EndFor

  ; Positions where there are good pixels
   qa_avg = (npix GT 0)
   wh = where(qa_avg EQ 1)

   ; Compute average for valid pixels
   avg = sum*0
   avg[wh] = sum[wh] / npix[wh]
```

I am also wondering what does += means?

Thank you!!!1

Subject: Re: Calculate the mean of many images
Posted by Craig Markwardt on Mon, 17 Nov 2008 07:13:27 GMT
View Forum Message <> Reply to Message

On Nov 16, 7:35 pm, Bulrush <Wasit.Weat...@gmail.com> wrote:
...
>
> Seems like I got lost between the codes. the mean I got with avg[wh]
> is all zero. Here is my code. Thanks for your help.
...
> mask = (QA EQ 0.00) AND (finite(Data) EQ 1) OR (QA EQ 1.00) AND
> (finite(Data) EQ 1)

So, does MASK select the pixels you want?

Also, you'd better make the order of operations explicit with
parentheses. You may or may not be getting what you expect.

> I am also wondering what does += means?

A += B ; is the same as
A = A + B

Craig

Subject: Re: Calculate the mean of many images
Posted by Jeremy Bailin on Mon, 17 Nov 2008 14:35:01 GMT
View Forum Message <> Reply to Message

On Nov 16, 9:57 am, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Nov 16, 1:16 am, Bulrush <Wasit.Weat...@gmail.com> wrote:
>
>
>
>> Hello,
>> I know this topic has been posted several times. But I could not find
>> my answer from these posted.
>> My issue is: I have many images 2 bands in each, one image is QA image
>> and the other one is data.
>> I need to calculate the mean of good pixels. Let's say QA image tells
>> me the location of good pixels, e.g. 1 for good pixels, and other for
>> bad. There are also NaN values. So, if the pixels are "good" in 7
>> images out of ten, then

>> (pixel1+pixel2...+Pixel7) /7
>
>> How can I do that?
>
>> Thanks any advice!
>
>> Elkunn
>
> I'm going to assume that qa and image are both nx x ny x nimage
> arrays, and qa is 1 for good, 0 for bad.
>
> ngoodpix = total(qa, 3)
> imagesum = total(image * qa, 3)
> meanimage = imagesum/ngoodpix
>
> -Jeremy.

Sorry, that doesn't deal with the NaNs properly... and you said that
qa can have other values than 1? How about this:

ngoodpix = total(qa ne 0, 3)
imagesum = total(image*finite(image)*(qa ne 0), 3)
meanimage = imagesum/ngoodpix

-Jeremy.

Subject: Re: Calculate the mean of many images
Posted by raghuram on Mon, 17 Nov 2008 16:38:14 GMT
View Forum Message <> Reply to Message

On Nov 16, 12:08 am, greg.a...@googlemail.com wrote:
> How about:
>
> q=where(QA eq 1, n)
> mean=total(image[q])/n
>
> cheers,
> Greg

To add to Greg's solution- For the NaN, you could use mean=total(image
[q]/n,/nan)
That should work.
