Subject: Re: divide by zero problems
Posted by korpela on Tue, 11 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3737C79C.372FA82E@aims.gov.au>,
Mark Rehbein <mrehbein@aims.gov.au> wrote:
> composite=sum image/ sum mask.
>
> sometimes my images are very cloudy and as such my "sum mask" has zero's
> in it and likewise for my "sum image". So I get divide by zero's and
> NaN's in my resultant composite image. In fact I get whole image of
> NaN's in my composite even though my sum image and sum mask have decent
> values in them.

Rather than using 0 in the mask image you could use a small non-zero value
like 1e-6.

Another option is after the division you can set the NaN values to another
value. The following statement will set NaN pixels in an image to 0.

image(where(image ne image))=0

Eric
--
Eric Korpela | An object at rest can never be
korpela@ssl.berkeley.edu | stopped.
Click for home page.

Subject: Re: divide by zero problems
Posted by davidf on Tue, 11 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Mark Rehbein (mrehbein@aims.gov.au) writes:

> I'm having problems with floating point divide by zero's. The
> application I'm using IDL for is image composition and masking.
> Now to composite or average the images I do the following:
>
> composite=sum image/ sum mask.
>
> sometimes my images are very cloudy and as such my "sum mask" has zero's
> in it and likewise for my "sum image". So I get divide by zero's and
> NaN's in my resultant composite image. In fact I get whole image of
> NaN's in my composite even though my sum image and sum mask have decent
> values in them.

Usually, something like this works:

composite = sum_image / (sum_mask > 10E-6)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: divide by zero problems
Posted by edward.s.meinel on Thu, 13 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

C'mon guys, this is IDL not C...

Mark Rehbein <mrehbein@aims.gov.au> wrote:

> Basically, my images have cloud pixels that I have changed to a zero
> value.

OK so far...

> I also have a mask of the image that has values 1 for good pixel
> and 0 for bad pixels (cloud).

So you have new_image = image*mask

> I then add all my images together to get
> a "sum image" and add all my masks together to get a "sum mask".

Oh, then its

```
FOR i = 0, number_of_images-1 DO BEGIN
  image_ sum = image_sum + image[*, *, i]*mask[*, *, i]
  mask_sum   = mask_sum + mask[*, *, i]
END
```

> Now to composite or average the images I do the following:
>
> composite=sum image/ sum mask.

i.e., you want to normalize the result. The you should use
something like:

composite = image_sum/(mask_sum + (mask_sum EQ 0))

> In fact I get whole image of
> NaN's in my composite even though my sum image and sum mask have
> decent values in them.

What is the TYPE of image? How many images? image_sum should be declared at least one TYPE higher than image (if image is BYTE then image_sum should be INTEGER; if image is INTEGER then image_sum should be LONG; etc.). If there are more than 256 images then mask_sum should be declared INTEGER.

Ed Meinel

Subject: Re: divide by zero problems
Posted by <asowter on Thu, 13 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I also had a similar problem but I used the FINITE function to test the result. FINITE(variable) returns false if you have a NaN and true otherwise - it's a nice test and allows you to skip or assign another value based on the input values (say if you know that there a tendency to a limit).

Andy
