

---

Subject: Re: Help on comparing 2 arrays  
Posted by [rogass](#) on Mon, 26 Apr 2010 14:52:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 26 Apr., 11:29, Dave Poreh <d.po...@gmail.com> wrote:

> Folks  
> I am trying to compare on ground (on sea!) laser data with MERIS data  
> for chlorophyll. Actually I have 2 arrays L[lat1, long1, c1] for on  
> ground measurements (with 400 meters resolution) and C[lat2, long2,  
> c2] for satellite data. What I want is this: for each pixel of C  
> (satellite data) extract data from array L that dropped inside of this  
> pixel. For instance for some pixels I have 3 or 4 data from L or  
> whatever. Does anyone have some good idea how to do this?  
> Any help highly appreciated.  
> Cheers  
> Dave

Hi Dave,  
maybe you can do this in the following way (as far as I understood  
what you want to do):

1. make a large array with a cell size of 10m covering the maximum spatial dimensions of the largest array
2. make the array 3D
3. fill in 2D layer 1 your `congrid(finite(L1),newsiz)`, in layer2 your `congrid(finite(L2),newsiz)` and in layer3 your `congrid(finite(C),newsiz)`, whereas in each layer the cells are covered according the spatial dimensions of the input - so for one MERIS cell 26x30 cells must contain a 1
4. ask for each 26x30 cell by `where(total(array,3) > 1)` for the related L1 or L2 measurements
5. rewrite the code to avoid for loops :)

I hope it helps somehow

Cheers

CR

---

---

Subject: Re: Help on comparing 2 arrays  
Posted by [Aram Panasenco](#) on Tue, 27 Apr 2010 02:48:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dave Poreh wrote:

> Folks  
> I am trying to compare on ground (on sea!) laser data with MERIS data  
> for chlorophyll. Actually I have 2 arrays L[lat1, long1, c1] for on

> ground measurements (with 400 meters resolution) and C[lat2, long2,  
> c2] for satellite data. What I want is this: for each pixel of C  
> (satellite data) extract data from array L that dropped inside of this  
> pixel. For instance for some pixels I have 3 or 4 data from L or  
> whatever. Does anyone have some good idea how to do this?  
> Any help highly appreciated.  
> Cheers  
> Dave

Hey Dave,

Are the latitude-longitude arrays in integer or floating-point format?  
Either way, you might want to specify an extraction radius - how close  
to each other do two points have to be to count them as the same point?  
I would define DELTA\_LAT and DELTA\_LON constants for that at the start  
of the routine.

Note: I see your arrays are in the format [latitude,longitude], so I  
will stick with that, but the normal convention is obviously  
[longitude,latitude]

I would then sort the L array in ascending latitude and ascending  
longitude orders:

```
sortL_Lat = sort(L[0,*])  
sortL_Lon = sort(L[1,*])
```

Then run a for-loop for every point in C and determine the indices of L  
where the point falls within the latitude range AND the longitude range.  
That gives you the indices of where to extract your L data for each  
point and do whatever you want with it.

If the radius calculations have to be a little more precise than that,  
you can use the MAP\_2POINTS routine on the points obtained using  
delta-longitude and delta-latitude comparison, and throw away the ones  
that are farther than some radian value away. Don't forget that the  
MAP\_2POINTS routine accepts data in the longitude,latitude format as  
opposed to your data format! ;)

Good Luck!

~Aram Panasenco

---

Subject: Re: Help on comparing 2 arrays  
Posted by [d.poreh](#) on Tue, 27 Apr 2010 09:13:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Apr 26, 7:48 pm, Aram Panasenco <panasenco...@gmail.com> wrote:

> Dave Poreh wrote:

>> Folks

>> I am trying to compare on ground (on sea!) laser data with MERIS data

>> for chlorophyll. Actually I have 2 arrays L[lat1, long1, c1] for on

>> ground measurements (with 400 meters resolution) and C[lat2, long2,

>> c2] for satellite data. What I want is this: for each pixel of C

>> (satellite data) extract data from array L that dropped inside of this

>> pixel. For instance for some pixels I have 3 or 4 data from L or

>> whatever. Does anyone have some good idea how to do this?

>> Any help highly appreciated.

>> Cheers

>> Dave

>

> Hey Dave,

>

> Are the latitude-longitude arrays in integer or floating-point format?

> Either way, you might want to specify an extraction radius - how close

> to each other do two points have to be to count them as the same point?

> I would define DELTA\_LAT and DELTA\_LON constants for that at the start

> of the routine.

>

> Note: I see your arrays are in the format [latitude,longitude], so I

> will stick with that, but the normal convention is obviously

> [longitude,latitude]

>

> I would then sort the L array in ascending latitude and ascending

> longitude orders:

>

> sortL\_Lat = sort(L[0,\*])

> sortL\_Lon = sort(L[1,\*])

>

> Then run a for-loop for every point in C and determine the indices of L

> where the point falls within the latitude range AND the longitude range.

> That gives you the indices of where to extract your L data for each

> point and do whatever you want with it.

>

> If the radius calculations have to be a little more precise than that,

> you can use the MAP\_2POINTS routine on the points obtained using

> delta-longitude and delta-latitude comparison, and throw away the ones

> that are farther than some radian value away. Don't forget that the

> MAP\_2POINTS routine accepts data in the longitude,latitude format as

> opposed to your data format! ;)

>

> Good Luck!

>

> ~Aram Panasenco

Thanks Guys

I did apply both way and they are brilliant. Chris's way is so fast because there is no for loops. Actually what I have done until now is extracting the satellite cells that ship went along (because my L array in L[lat1, long1, c1] format), but sounds like it is better to do comparison with the original scenes as Chris suggested. I give it a try with that way also.

Cheers

---