

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [David Fanning](#) on Mon, 12 Dec 2011 02:29:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Jasdeep Anand writes:

- > I have some satellite data I'd like to bin to a high-resolution 2D
- > grid for plotting and/or other analytical purposes. Each data point
- > that I have has a corresponding latitude and longitude of the centre
- > of the original pixel that the data was originally recorded from. The
- > grid I'm trying to assign this data to is of a much finer resolution
- > than what the data was taken from, so several data points may be
- > assigned to the same pixel, in which case I guess an average data
- > value will need to be assigned instead. To further complicate matters,
- > the data I want to bin is a global dataset of ~1500000 individual
- > points, spread over a number of ASCII files.
- >
- > Are there any routines that can handle having input data this large?
- > Ideally I'd like to incorporate the binning process into the same loop
- > that extracts and reads the data from each file, but I think functions
- > like GRIDDATA require having all the data points to be gridded
- > available already when calling them. Also, are there any general
- > "housekeeping" tips that anyone can tell me about handling such data?
- > I'm still quite new to this, and would appreciate any pointers you all
- > could give me!

If this were me, I wouldn't think about doing this in IDL at all. I'd spend all my time trying to convince some hapless graduate student that he would be famous if he would write a C program to do this. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Jasdeep Anand](#) on Mon, 12 Dec 2011 09:51:57 GMT

---

On Dec 12, 2:29 am, David Fanning <n...@dfanning.com> wrote:

> Jasdeep Anand writes:

>> I have some satellite data I'd like to bin to a high-resolution 2D  
>> grid for plotting and/or other analytical purposes. Each data point  
>> that I have has a corresponding latitude and longitude of the centre  
>> of the original pixel that the data was originally recorded from. The  
>> grid I'm trying to assign this data to is of a much finer resolution  
>> than what the data was taken from, so several data points may be  
>> assigned to the same pixel, in which case I guess an average data  
>> value will need to be assigned instead. To further complicate matters,  
>> the data I want to bin is a global dataset of ~1500000 individual  
>> points, spread over a number of ASCII files.

>

>> Are there any routines that can handle having input data this large?  
>> Ideally I'd like to incorporate the binning process into the same loop  
>> that extracts and reads the data from each file, but I think functions  
>> like GRIDDATA require having all the data points to be gridded  
>> available already when calling them. Also, are there any general  
>> "housekeeping" tips that anyone can tell me about handling such data?  
>> I'm still quite new to this, and would appreciate any pointers you all  
>> could give me!

>

> If this were me, I wouldn't think about doing this in  
> IDL at all. I'd spend all my time trying to convince some  
> hapless graduate student that he would be famous if he would  
> write a C program to do this. :-)

>

> Cheers,

>

> David

>

> --

> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming:<http://www.idlcoyote.com/>

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")- Hide quoted text -

>

> - Show quoted text -

David,

Considering that I am a hapless grad student, would fame and fortune  
really await me if I can crack this? ;)

In all seriousness though, would routines like GRIDDATA, TRIGRID, etc  
break down for such a large input? How do people in general handle  
analysing such large datasets? Again, I'd be grateful for any advice

you all could give me.

Thanks again,

Jasdeep.

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Fabzou](#) on Mon, 12 Dec 2011 10:34:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

> In all seriousness though, would routines like GRIDDATA, TRIGRID, etc  
> break down for such a large input?

Well, I am not very familiar with GRIDDATA but 1500000 points is not so large.

It not difficult to find out. It mostly depends on your available memory, but it seems alright. IDL is just not very very fast. If you have to do it many times, that's maybe not the best tool for it...

```
n = 1500000L
lons = Scale_Vector(RANDOMU(seed, n), -180., 180)
lats = Scale_Vector(RANDOMU(seed, n), -90., 90.)
TRIANGULATE, lons, lats, Triangles
data = FLTARR(n)
out = GRIDDATA(lons, lats, data, $
  START=[0.05D,0.05D], DIMENSION=[3600,1800], DELTA=[0.1D,0.1D],$
  TRIANGLES=triangles, /NEAREST_NEIGHBOR)
```

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Jasdeep Anand](#) on Mon, 12 Dec 2011 10:41:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Dec 12, 10:34 am, Fabzou <fabien.mauss...@tu-berlin.de> wrote:

>> In all seriousness though, would routines like GRIDDATA, TRIGRID, etc  
>> break down for such a large input?

>

> Well, I am not very familiar with GRIDDATA but 1500000 points is not so  
> large.

>

> It not difficult to find out. It mostly depends on your available  
> memory, but it seems alright. IDL is just not very very fast. If you  
> have to do it many times, that's maybe not the best tool for it...

```
>
> n = 1500000L
> lons = Scale_Vector(RANDOMU(seed, n), -180., 180)
> lats = Scale_Vector(RANDOMU(seed, n), -90., 90.)
> TRIANGULATE, lons, lats, Triangles
> data = FLTARR(n)
> out = GRIDDATA(lons, lats, data, $
>   START=[0.05D,0.05D], DIMENSION=[3600,1800], DELTA=[0.1D,0.1D],$
>   TRIANGLES=triangles, /NEAREST_NEIGHBOR)
```

Thanks Fabozu! I'm coding my own attempt that this as we speak.

I've noticed from the few examples I've seen from the web that both TRIGRID and GRIDDATA can be used for this problem. How do both routines differ from each other, and when should either one be used? Again, I'm grateful for any advice you all can give me - I'm still learning!

---

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Fabzou](#) on Mon, 12 Dec 2011 12:08:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

```
> I've noticed from the few examples I've seen from the web that both
> TRIGRID and GRIDDATA can be used for this problem. How do both
> routines differ from each other, and when should either one be used?
```

Again, I am not the expert here, somebody might jump in to add some info, but GRIDDATA has more interpolation options than TRIGRID does.

The choice of the interpolation scheme is highly dependent of the type of data you are willing to interpolate and their own spatial validity. What kind of data do you want to interpolate?

---

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Jasdeep Anand](#) on Mon, 12 Dec 2011 13:13:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Dec 12, 12:08 pm, Fabzou <fabien.mauss...@tu-berlin.de> wrote:

```
>> I've noticed from the few examples I've seen from the web that both
>> TRIGRID and GRIDDATA can be used for this problem. How do both
>> routines differ from each other, and when should either one be used?
>
> Again, I am not the expert here, somebody might jump in to add some
```

- > info, but GRIDDATA has more interpolation options than TRIGRID does.
- >
- > The choice of the interpolation scheme is highly dependent of the type
- > of data you are willing to interpolate and their own spatial validity.
- > What kind of data do you want to interpolate?

Fabozu,

I'm currently investigating pollution trends (namely NO2) over large areas using data from the OMI (AURA) instrument. The pixels at nadir have a size of 13 x 24 km, but I want to regrid those to a much higher resolution. I would ideally like an uninterrupted dataset without missing pixels so I can do a Fourier analysis around certain latitude bands, which is why I'm looking to interpolate them in the first place.

---

Subject: Re: The best way to bin data to a grid? (may not be an IDL-specific question)

Posted by [Fabzou](#) on Tue, 13 Dec 2011 09:58:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This kind of data is probably valid for the whole pixel, so it is probably not the best idea to use a much higher resolution... Nearest neighbour is probably the best way to stay close from "reality" in this case

On 12/12/2011 02:13 PM, Jasdeep Anand wrote:

- > On Dec 12, 12:08 pm, Fabzou<fabien.mauss...@tu-berlin.de> wrote:
- >>> I've noticed from the few examples I've seen from the web that both
- >>> TRIGRID and GRIDDATA can be used for this problem. How do both
- >>> routines differ from each other, and when should either one be used?
- >>
- >> Again, I am not the expert here, somebody might jump in to add some
- >> info, but GRIDDATA has more interpolation options than TRIGRID does.
- >>
- >> The choice of the interpolation scheme is highly dependent of the type
- >> of data you are willing to interpolate and their own spatial validity.
- >> What kind of data do you want to interpolate?
- >
- > Fabozu,
- >
- > I'm currently investigating pollution trends (namely NO2) over large
- > areas using data from the OMI (AURA) instrument. The pixels at nadir
- > have a size of 13 x 24 km, but I want to regrid those to a much higher
- > resolution. I would ideally like an uninterrupted dataset without
- > missing pixels so I can do a Fourier analysis around certain latitude
- > bands, which is why I'm looking to interpolate them in the first

> place.

---