Subject: Interpolation question
Posted by rkj on Wed, 25 Jul 2001 18:18:30 GMT
View Forum Message <> Reply to Message

I have an image containg some bad values. I would like
to replace these points with the average value of their
neighbors. Is there an easy way to do this without loops?

For instance, a 3x3 array as follows:

1 1 1
1 0 1
1 1 1

would become

1 1 1
1 1 1
1 1 1.

I can't get the boundary conditions to work correctly when
I use CONVOL. It either zeros the edges or does not process
them.

For instance,

1 1 1
0 1 1
1 1 0

should be

1 1 1
1 1 1
1 1 1

as well.

Kyle J.

Subject: Re: interpolation question
Posted by peter.albert@gmx.de on Thu, 20 Apr 2006 08:20:16 GMT
View Forum Message <> Reply to Message

Hi Chen,

I haven't tested this extensively, but I would try

nn = y[round(interpol(indgen(n_elements(x)),x, u))]

where "x" and "y" are the, well, x and y values and "u" is the vector
with x-values for which the nearest neighbours are to be found.

What goes on?

The interpol command interpolates the indices of the x and y vectors to
the new locations, round actually finds the nearest neighbour in terms
of index, and then we just assign y with those indices.

Regards,

Peter

Subject: Re: interpolation question
Posted by chen123.dian on Thu, 20 Apr 2006 14:50:38 GMT
View Forum Message <> Reply to Message

Thanks, Peter,

Your code works well for 1-D data. It will be nice to extend it to 2-D
data.

Regards,

Chen

Peter Albert wrote:
> Hi Chen,
>
> I haven't tested this extensively, but I would try
>
> nn = y[round(interpol(indgen(n_elements(x)),x, u))]
>
> where "x" and "y" are the, well, x and y values and "u" is the vector
> with x-values for which the nearest neighbours are to be found.
>
> What goes on?
>
> The interpol command interpolates the indices of the x and y vectors to
> the new locations, round actually finds the nearest neighbour in terms
> of index, and then we just assign y with those indices.
>
> Regards,

>
> Peter

Subject: Re: interpolation question
Posted by peter.albert@gmx.de on Thu, 20 Apr 2006 15:47:39 GMT
View Forum Message <> Reply to Message

Hi Chen,

if your input array is 2D, things are a bit more tricky. By brue force,
you have to calculate all differencences between all grod points and
all "nearest neighbour" points, which, given sufficiently large arrays,
is either very slow or very memory consuing, or both. But luckily,
astronomers seem to have a need for this, therefore it's all done,
after a discussion in this group, David has provided two routines and
the way how to get there at

http://www.dfanning.com/code_tips/slowloops.html

Regards,

  Peter

Subject: Re: interpolation question
Posted by Mark Hadfield on Thu, 20 Apr 2006 19:26:58 GMT
View Forum Message <> Reply to Message

chen123.dian@gmail.com wrote:
> Thanks, Peter,
>
> Your code works well for 1-D data. It will be nice to extend it to 2-D
> data.

The key to solving your problem is locating the points of your output
grid in the "index space" of your input grid. There are routines to do
this in the Motley library:

  http://www.dfanning.com/hadfield/idl/README.html

They are called MGH_LOCATE and MGH_LOCATE2, for 1D and 2D respectively.
(The 1D version uses the INTERPOL trick suggested by Peter.) Their use
was discussed on the group recently in a couple of threads entitled
"Interpolating a regular grid" and "matching 2 grids". See

  http://tinyurl.com/r9t5s

http://tinyurl.com/fherp

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: interpolation question
Posted by chen123.dian on Mon, 24 Apr 2006 18:35:27 GMT
View Forum Message <> Reply to Message

Hi,

Thanks so much for all suggestions here.

I am so wonder that why IDL has no simple function like MATLAB's 'interp2'. Another problem is for value_locate. Some suggestions mentioned to use value_locate. Here is a example to show my problem.

```
IDL> vec = [2.0, 5.0, 8.0, 10.0]
IDL> print, vec
     2.00000      5.00000      8.00000      10.0000
IDL> loc = VALUE_LOCATE(vec, [0.0, 4.5, 5.0, 6.0, 12.0])
IDL> print, loc
        -1           0           1           1           3
```

We can see the value of "4.5" corresponds to index location of "0". Actually, the value of "4.5" should correspond to index location of "1" because the value of "4.5" is closer to the value of "5.0" having index location of "1".

Thanks.

Regards,

Chen

Subject: Re: interpolation question
Posted by Jo Klein on Mon, 24 Apr 2006 19:04:55 GMT
View Forum Message <> Reply to Message

Hi Chen,
VALUE_LOCATE works differently: It returns the location as the interval in which your value is to be found, vec[location]<=value<vec[location+1]
It's not designed to be a nearest-neighbour-style routine. Granted, the

function name is a bit misleading ...
Cheers,
Jo

chen123.dian@gmail.com wrote:
> Hi,
>
> Thanks so much for all suggestions here.
>
> I am so wonder that why IDL has no simple function like MATLAB's
> 'interp2'. Another problem is for value_locate. Some suggestions
> mentioned to use value_locate. Here is a example to show my problem.
>
> IDL> vec = [2.0, 5.0, 8.0, 10.0]
> IDL> print, vec
> 2.00000 5.00000 8.00000 10.0000
> IDL> loc = VALUE_LOCATE(vec, [0.0, 4.5, 5.0, 6.0, 12.0])
> IDL> print, loc
> -1 0 1 1 3
>
> We can see the value of "4.5" corresponds to index location of "0".
> Actually, the value of "4.5" should correspond to index location of "1"
> because the value of "4.5" is closer to the value of "5.0" having index
> location of "1".
>
> Thanks.
>
> Regards,
>
> Chen
>

Subject: Re: interpolation question
Posted by Paul Van Delst[1] on Mon, 24 Apr 2006 19:09:27 GMT
View Forum Message <> Reply to Message

chen123.dian@gmail.com wrote:
> Hi,
>
> Thanks so much for all suggestions here.
>
> I am so wonder that why IDL has no simple function like MATLAB's
> 'interp2'. Another problem is for value_locate. Some suggestions
> mentioned to use value_locate. Here is a example to show my problem.
>
> IDL> vec = [2.0, 5.0, 8.0, 10.0]
> IDL> print, vec

> 2.00000 5.00000 8.00000 10.0000
> IDL> loc = VALUE_LOCATE(vec, [0.0, 4.5, 5.0, 6.0, 12.0])
> IDL> print, loc
> -1 0 1 1 3
>
> We can see the value of "4.5" corresponds to index location of "0".
> Actually, the value of "4.5" should correspond to index location of "1"
> because the value of "4.5" is closer to the value of "5.0" having index
> location of "1".

From the documentation:

<quote>
Result = VALUE_LOCATE (Vector, Value [, /L64])
Return Value
Each return value, Result [i], is an index, j, into Vector, corresponding to the interval into which the given Value [i] falls. The returned values are in the range -1 £ j £ N-1, where N is the number of elements in the input vector
</quote>

So the results you've got seem to be correct.

paulv

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx
Ph: (301)763-8000 x7748
Fax:(301)763-8545

Subject: Re: interpolation question
Posted by news.verizon.net on Mon, 24 Apr 2006 19:47:50 GMT
View Forum Message <> Reply to Message

> I am so wonder that why IDL has no simple function like MATLAB's
> 'interp2'.

I agree with you that IDL appears deficient to MATLAB in providing an easy and consistent set of interpolation routines. First of all, while IDL has interpol.pro for 1d interpolation, there is no equivalent function for 2-d interpolation. (bilinear.pro and interpolate require you to supply indicies). And while interpol.pro provides several interpolation methods, it doesn't include the simplest (nearest neighbor), though this would be easy to add (as in http://idlastro.gsfc.nasa.gov/ftp/pro/math/linterp.pro). It makes

much more sense to have functions interp1d and interp2d, each with a
variety of interpolation methods available.

> Another problem is for value_locate. Some suggestions
> mentioned to use value_locate. Here is a example to show my problem.
>
> IDL> vec = [2.0, 5.0, 8.0, 10.0]
> IDL> print, vec
> 2.00000 5.00000 8.00000 10.0000
> IDL> loc = VALUE_LOCATE(vec, [0.0, 4.5, 5.0, 6.0, 12.])
DL> print, loc
  -1 0 1 1 3

VALUE_LOCATE is doing what it says it does -- returning a value j such
that
vec[j] < x < vec[j+1]. I don't think anyone suggested that
VALUE_LOCATE can give you the answer by itself, but both JD Smith and
the archive posting from David Fanning (http://tinyurl.com/r9t5s)
showed how you could use VALUE_LOCATE to get
the index of the nearest value. In this case

loc = round(loc+(x-vec[loc]) / (vec[loc+1]-vec[loc]))

(Actually one should first check that 0 < loc < N_Elements(vec)-1, as
in
http://idlastro.gsfc.nasa.gov/ftp/pro/math/tabinv.pro)

--Wayne

Subject: Re: Interpolation question
Posted by JD Smith on Wed, 02 Aug 2006 16:31:24 GMT
View Forum Message <> Reply to Message

On Wed, 02 Aug 2006 11:21:55 -0500, Mike Wallace wrote:

> I have an array that I need to interpolate and would like to find a
> semi-efficient way to do it. Well, almost anything would be more
> efficient than what I'm currently doing.
>
> I have an array of data and a corresponding array of times when the data
> was taken. I have a third array that represents the times that I want
> to calculate the interpolation. How can I easily (and efficiently)
> calculate the data points corresponding to that array?
>
>
> For example...

>
> data = [12, 6, 1, 4, 8, 8, 10]
> time = [0, 1, 4, 7, 8, 11, 14]
>
> Now, I say that I wanted to interpolate the data array but only
> calculate the interpolation for the time values in some new array...
>
> new_time = [1, 4, 5, 8, 9, 12]

 new_data=interpol(float(data),time,new_time)

The float is needed or all calcs are done in integer space (probably not
what you want).

JD

Subject: Re: Interpolation question
Posted by mchinand on Wed, 02 Aug 2006 16:37:09 GMT
View Forum Message <> Reply to Message

In article <12d1k3ah497rj6c@corp.supernews.com>,
Mike Wallace <mwallace.no.spam.please@swri.edu.invalid> wrote:
> I have an array that I need to interpolate and would like to find a
> semi-efficient way to do it. Well, almost anything would be more
> efficient than what I'm currently doing.
>
> I have an array of data and a corresponding array of times when the data
> was taken. I have a third array that represents the times that I want
> to calculate the interpolation. How can I easily (and efficiently)
> calculate the data points corresponding to that array?
>
>
> For example...
>
> data = [12, 6, 1, 4, 8, 8, 10]
> time = [0, 1, 4, 7, 8, 11, 14]
>
> Now, I say that I wanted to interpolate the data array but only
> calculate the interpolation for the time values in some new array...
>
> new_time = [1, 4, 5, 8, 9, 12]
>

Check out INTERPOL:

IDL> new_data=interpol(data,time,new_time)
IDL> print, new_data

6.00000 1.00000 2.00000 8.00000 8.00000 8.66667

I changed the data array to float, otherwise the resulting new_data array is Integer type which you may or may not want. INTERPOL has different interpolation types to choose from, linear, quadratic,
or spline.

Hope that helps,

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago

Subject: Re: Interpolation question
Posted by Mike Wallace on Wed, 02 Aug 2006 20:34:56 GMT
View Forum Message <> Reply to Message

> new_data=interpol(float(data),time,new_time)
>
> The float is needed or all calcs are done in integer space (probably not
> what you want).

This is one of the things that I love and hate about IDL. I love that IDL usually has a function that does what I want to do. I hate that I can never figure out which function is named. You should have seen how long it took me to find n_elements() when I was first learning the language. Thanks again guys.

-Mike

Subject: Re: Interpolation question
Posted by greg michael on Thu, 03 Aug 2006 11:04:24 GMT
View Forum Message <> Reply to Message

ha! I know what you mean. There was something yesterday about a function called file_lines - wish I'd discovered that years ago...

Greg

Subject: Re: Interpolation question
Posted by mchinand on Thu, 03 Aug 2006 16:01:32 GMT
View Forum Message <> Reply to Message

In article <1154603064.459904.56640@b28g2000cwb.googlegroups.com>,
greg michael <greg.michael@gmail.com> wrote:
> ha! I know what you mean. There was something yesterday about a
> function called file_lines - wish I'd discovered that years ago...
>
> Greg
>

In the IDL help, there's a functional list of functions/procedures, that lists them by category (graphics, array manipulation, etc.) along with a one line description. The old PDF version of the help made it easier to browse the list than the new help system, but the new system is still very handy.

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago

Subject: file_lines and expand path
Posted by FL on Fri, 04 Aug 2006 15:23:37 GMT
View Forum Message <> Reply to Message

Hi,

I am implementing file_lines in FL and was wondering what to return when the input path expands to multiple files. The IDL reference guide says nothing. I have run IDL and got curious result:

```
IDL> help, file_lines('*.pro')
<Expression>    LONG64    =                15
```

I have 448 .pro files in the current directory, with a total of 10k lines. IDL seems to be randomly picking a single file and reporting its length (probably the first match in readdir()).

What do you expect to get?

Possibilities:

1. never expand (but why do we have NOEXPAND then?)

2. give an error message if the expansion results in more than one file
3. return the total length of files
4. return the total length of files if a new keyword TOTAL was set
5. do as IDL, pick a single file silently

thanks,
lajos
