Subject: Re: 3D congrid without interpolation
Posted by David Fanning on Thu, 12 Apr 2007 21:19:45 GMT
View Forum Message <> Reply to Message

mxhamidi@gmail.com writes:

> Is it possible to resize a 3D array in idl without being forced to
> interpolate? I am using discrete values to represent different
> information and interpolation alters those values. I can't use rebin
> since I'd like to change the image size by a non-integral factor.

Humm. Hard for me to imagine what you are using to
do this that is interpolating anything for you.
CONGRID is normally used, but that won't interpolate
unless you explicitly tell it to.

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: 3D congrid without interpolation
Posted by Michael Galloy on Thu, 12 Apr 2007 22:40:07 GMT
View Forum Message <> Reply to Message

On Apr 12, 3:19 pm, David Fanning <n...@dfanning.com> wrote:
> Humm. Hard for me to imagine what you are using to
> do this that is interpolating anything for you.
> CONGRID is normally used, but that won't interpolate
> unless you explicitly tell it to.

CONGRID interpolates 3-dimensional arrays by default. From the online
help for the INTERP keyword for CONGRID:

INTERP
Set this keyword to force CONGRID to use linear interpolation when
resizing a 1- or 2-dimensional array. CONGRID automatically uses
linear interpolation if the input array is 3-dimensional. When the
input array is 1- or 2-dimensional, the default is to employ nearest-
neighbor sampling.

Mike
--

Subject: Re: 3D congrid without interpolation
Posted by David Fanning on Fri, 13 Apr 2007 00:30:24 GMT
View Forum Message <> Reply to Message

mgalloy@gmail.com writes:

> CONGRID automatically uses
> linear interpolation if the input array is 3-dimensional.

Aaaugghh! I *always* forget that! :-(

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: 3D congrid without interpolation
Posted by Qing on Fri, 13 Apr 2007 02:45:27 GMT
View Forum Message <> Reply to Message

On Apr 13, 10:30 am, David Fanning <n...@dfanning.com> wrote:
> mgal...@gmail.com writes:
>> CONGRID automatically uses
>> linear interpolation if the input array is 3-dimensional.
>
> Aaaugghh! I *always* forget that! :-(
>

What about to do it in two steps:
(1) loop through the 3rd dimension while using CONGRID in 2D mode;
(2) if you need to resize the 3rd dimension, then loop through 2nd
dimension with 2D CONGRID again.
Qing :-)

Subject: Re: 3D congrid without interpolation
Posted by Mike[2] on Fri, 13 Apr 2007 15:21:46 GMT

On Apr 12, 6:40 pm, "mgal...@gmail.com" <mgal...@gmail.com> wrote:
> CONGRID interpolates 3-dimensional arrays by default.

If you are really looking for a method that requires no interpolation, you will have to choose your new array to overlap a subset of the array indexes of the original data. I suspect you might really be thinking about nearest neighbor interpolation. If so, you can do that by

1 - calculating the [x,y,z] coordinates at which I want to evaluate my data.

2 - round the coordinates

3 - interpolate a new array with interpolate(data,x,y,z)

This can be a memory hog for large arrays since you need 4 arrays for each point in the new array. In the case where this leads to lots of swapping, I usually do it slice by slice.

Mike

Subject: Re: 3D congrid without interpolation
Posted by JD Smith on Fri, 13 Apr 2007 22:57:37 GMT

On Thu, 12 Apr 2007 15:40:07 -0700, mgalloy@gmail.com wrote:

> On Apr 12, 3:19 pm, David Fanning <n...@dfanning.com> wrote:
>> Humm. Hard for me to imagine what you are using to
>> do this that is interpolating anything for you.
>> CONGRID is normally used, but that won't interpolate
>> unless you explicitly tell it to.
>
> CONGRID interpolates 3-dimensional arrays by default. From the online
> help for the INTERP keyword for CONGRID:
>
> INTERP
> Set this keyword to force CONGRID to use linear interpolation when
> resizing a 1- or 2-dimensional array. CONGRID automatically uses
> linear interpolation if the input array is 3-dimensional. When the
> input array is 1- or 2-dimensional, the default is to employ nearest-
> neighbor sampling.

How is "nearest neighbor sampling" not interpolation? Does it

explicitly avoid knowledge of how the new array cell is positioned
w.r.t. the old one, and simply grab averages of nearby neighbors? Why
would this ever be preferable to a linear interpolation?

BTW, there has be a good deal of progress on interpolators, especially
for image data, which IDL hasn't taken advantage of. For example,
when downsizing, you need to take care to avoid moire artifacts and
aliasing.

Here's a classic comparison of various interpolators:

http://www.all-in-one.ee/~dersch/interpolator/interpolator.h tml

JD

Subject: Re: 3D congrid without interpolation
Posted by David Fanning on Sat, 14 Apr 2007 00:33:53 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> How is "nearest neighbor sampling" not interpolation? Does it
> explicitly avoid knowledge of how the new array cell is positioned
> w.r.t. the old one, and simply grab averages of nearby neighbors? Why
> would this ever be preferable to a linear interpolation?

I don't know. I guess it is preferable because it
doesn't add new numbers to your data. (I never really
thought about or cared how it was done, but I suppose
someone ought to.)

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: 3D congrid without interpolation
Posted by JD Smith on Sat, 14 Apr 2007 01:00:48 GMT
View Forum Message <> Reply to Message

On Fri, 13 Apr 2007 17:33:53 -0700, David Fanning wrote:

> JD Smith writes:
>
>> How is "nearest neighbor sampling" not interpolation? Does it
>> explicitly avoid knowledge of how the new array cell is positioned
>> w.r.t. the old one, and simply grab averages of nearby neighbors? Why
>> would this ever be preferable to a linear interpolation?
>
> I don't know. I guess it is preferable because it
> doesn't add new numbers to your data. (I never really
> thought about or cared how it was done, but I suppose
> someone ought to.)

Aha, well I guess it really does just pick the nearest neighboring cell,
so it is not interpolation (I presumed it was averaging over neighbors
without weighting).

```
IDL> a=findgen(5,5)
IDL> print,a
     0.00000     1.00000     2.00000     3.00000     4.00000
     5.00000     6.00000     7.00000     8.00000     9.00000
     10.0000     11.0000     12.0000     13.0000     14.0000
     15.0000     16.0000     17.0000     18.0000     19.0000
     20.0000     21.0000     22.0000     23.0000     24.0000
IDL> print,congrid(a,4,4)
     0.00000     1.00000     2.00000     3.00000
     5.00000     6.00000     7.00000     8.00000
     10.0000     11.0000     12.0000     13.0000
     15.0000     16.0000     17.0000     18.0000
IDL> print,congrid(a,4,4,/INTERP)
     0.00000     1.25000     2.50000     3.75000
     6.25000     7.50000     8.75000     10.0000
     12.5000     13.7500     15.0000     16.2500
     18.7500     20.0000     21.2500     22.5000
```

JD

Subject: Re: 3D congrid without interpolation
Posted by James Kuyper on Sat, 14 Apr 2007 17:59:26 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
> On Thu, 12 Apr 2007 15:40:07 -0700, mgalloy@gmail.com wrote:
>
>> On Apr 12, 3:19 pm, David Fanning <n...@dfanning.com> wrote:
>>> Humm. Hard for me to imagine what you are using to
>>> do this that is interpolating anything for you.

>>> CONGRID is normally used, but that won't interpolate
>>> unless you explicitly tell it to.
>>
>> CONGRID interpolates 3-dimensional arrays by default. From the online
>> help for the INTERP keyword for CONGRID:
>>
>> INTERP
>> Set this keyword to force CONGRID to use linear interpolation when
>> resizing a 1- or 2-dimensional array. CONGRID automatically uses
>> linear interpolation if the input array is 3-dimensional. When the
>> input array is 1- or 2-dimensional, the default is to employ nearest-
>> neighbor sampling.
>
> How is "nearest neighbor sampling" not interpolation? Does it
> explicitly avoid knowledge of how the new array cell is positioned
> w.r.t. the old one, and simply grab averages of nearby neighbors? ...

No. It grabs the value of the one nearest neighbor, with appropriate
rules for breaking ties. No averaging of any kind is done on that
value, which is why it's inappropriate to call this 'interpolation'.

> ... Why
> would this ever be preferable to a linear interpolation?

Well, for one thing, it's a lot faster.

However, another good reason is if you're re-binning categorical
data, where the codes representing each category are arbitrary, and
it's simply not meaningful to take the average of the category codes.
If category 1 means 'corn' and category 3 means 'wheat', you don't
want a thin barrier line of category 2 (meaning 'barley') to ocurr at
the boundaries between wheat fields and corn fields. when you rebin
your data. Nearest neighbor interpolation will always generate either
1 or 3 along that boundary.

A third case that I'm very familiar with is mainly of use for
debugging purposes. I'm responsible for programs which calibrate and
geolocate satellite images. Occasionally I want to create a plot with
a resolution much higher than the resolution of our images, using
nearest negghbor interpolation. For any given low-resolution pixel,
there's multiple high-resolution pixels for which it is the nearest
neighbor, and they all get assigned the same color. As a result, I can
very clearly where the boundaries are between the low-resolution
pixels. That helps me decide whether or not we've geolocated those
pixels correctly.

Subject: Re: 3D congrid without interpolation
Posted by mxhamidi on Mon, 16 Apr 2007 18:43:32 GMT
View Forum Message <> Reply to Message

On Apr 14, 12:59 pm, kuy...@wizard.net wrote:

> However, another good reason is if you're re-binning categorical
> data, where the codes representing each category are arbitrary, and
> it's simply not meaningful to take the average of the category codes.
> If category 1 means 'corn' and category 3 means 'wheat', you don't
> want a thin barrier line of category 2 (meaning 'barley') to ocurr at
> the boundaries between wheat fields and corn fields. when you rebin
> your data. Nearest neighbor interpolation will always generate either
> 1 or 3 along that boundary.

This is very similar to my concern. I have maps of brain activity with each value representing seeing a different angle in visual space. I need resize my brain activity map (64 x 64 x 30) to fit onto the anatomical image of the brain (256 x 256 x 256). With congrid (at least with 3D congrid) the label of each coordinate is altered making the resulting image uninterpretable. I think that I see if Qing's idea of doing two steps of 2-D congrid will resize the activity map without any averaging of the values.

Thanks for your replies.

Subject: Re: 3D congrid without interpolation
Posted by JD Smith on Mon, 23 Apr 2007 16:53:26 GMT
View Forum Message <> Reply to Message

On Sat, 14 Apr 2007 10:59:26 -0700, kuyper wrote:

> JD Smith wrote:
>> On Thu, 12 Apr 2007 15:40:07 -0700, mgalloy@gmail.com wrote:
>>
>>> On Apr 12, 3:19 pm, David Fanning <n...@dfanning.com> wrote:
>>>> Humm. Hard for me to imagine what you are using to
>>>> do this that is interpolating anything for you.
>>>> CONGRID is normally used, but that won't interpolate
>>>> unless you explicitly tell it to.
>>>
>>> CONGRID interpolates 3-dimensional arrays by default. From the online
>>> help for the INTERP keyword for CONGRID:
>>>
>>> INTERP
>>> Set this keyword to force CONGRID to use linear interpolation when
>>> resizing a 1- or 2-dimensional array. CONGRID automatically uses

>>> linear interpolation if the input array is 3-dimensional. When the
>>> input array is 1- or 2-dimensional, the default is to employ nearest-
>>> neighbor sampling.
>>
>> How is "nearest neighbor sampling" not interpolation? Does it
>> explicitly avoid knowledge of how the new array cell is positioned
>> w.r.t. the old one, and simply grab averages of nearby neighbors? ...
>
> No. It grabs the value of the one nearest neighbor, with appropriate
> rules for breaking ties. No averaging of any kind is done on that
> value, which is why it's inappropriate to call this 'interpolation'.

Right, David corrected me. I guess my brain was registering "nearest neighbor interpolation" with some weighted averaging.

>> ... Why
>> would this ever be preferable to a linear interpolation?
>
> Well, for one thing, it's a lot faster.
>
> However, another good reason is if you're re-binning categorical
> data, where the codes representing each category are arbitrary, and
> it's simply not meaningful to take the average of the category codes.
> If category 1 means 'corn' and category 3 means 'wheat', you don't
> want a thin barrier line of category 2 (meaning 'barley') to ocurr at
> the boundaries between wheat fields and corn fields. when you rebin
> your data. Nearest neighbor interpolation will always generate either
> 1 or 3 along that boundary.

Good example.

Thanks,

JD
