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.
