
Subject: interpolation for resizing

Posted by [bryan.s.hong](#) on Tue, 02 Dec 2008 22:29:39 GMT

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

I have a 100 X 100 spatial image to be resized to 1000*1000.

The data of this image is surface temperature and includes some part of ocean.

Because I want to exclude data of ocean area during interpolation, I'm trying to resize it using "interpolate".

But I cannot understand the IDL help pages for the 'interpolate'.

Could anyone help me with a simple command line for this job? Thanks.

- Bryan

Subject: Re: interpolation for resizing

Posted by [Jean H.](#) on Fri, 05 Dec 2008 16:04:50 GMT

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

> I guess you didn't read my question properly (although I maybe didn't
> have to ask it in this thread to avoid this, but I didn't want to make
> a new thread because of David's post several days ago). I'm not trying
> to resize anything. I just want the -999 or NaN (I make them with !
> Values.F_NAN option btw.) values to have a value in the b array that
> is calculated by interpolation (for instance: the IDW of the
> surroundig pixels in a 3x3 window).
>
> So sorry for the mixup and thanks if you guys still feel like finding
> a solution for this.

Hi,

ah, now you give more information... a 3*3 window, and IDW... well, the "do it yourself" solution is the easiest... You can put that in a function or else.

pro test_interpol

nx=10

ny=10

data = findgen(nx,ny)

data[2,2] = !VALUES.F_NAN

data[6,6] = !VALUES.F_NAN

;print,data

badIdx = where(finite(data) eq 0, countBad)

print,'Bad index: ', badIdx

```

;badNeighbors = rebin(badIdx, 8 ,countBad)
badNeighbors = transpose(rebin(badIdx, countBad,8))

;Get the index of the cells in the Moore neighborhood of each bad
;pixel (you deal with the pixels on the border...)
badNeighbors[0,*] -= 1
badNeighbors[1,*] += 1
badNeighbors[2,*] -= nx
badNeighbors[3,*] -= nx-1
badNeighbors[4,*] -= nx+1
badNeighbors[5,*] += nx
badNeighbors[6,*] += nx-1
badNeighbors[7,*] += nx+1

print, 'Neighbors of bad index: ',badNeighbors

goodValue = (data[badNeighbors[0,*]] + data[badNeighbors[1,*]] +$
data[badNeighbors[2,*]] + data[badNeighbors[3,*]]+$
data[badNeighbors[4,*]] + data[badNeighbors[5,*]] +$
data[badNeighbors[6,*]] +data[badNeighbors[7,*]])/8
;Adjuste the equation yourself.

print,'Interpolated values: ', goodValue

data[badIdx] = goodValue

print,'The new data: ', data
end

```

Jean

Subject: Re: interpolation for resizing

Posted by [ben.bighair](#) on Sun, 07 Dec 2008 16:57:20 GMT

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

On Dec 5, 11:04 am, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca> wrote:

```

>> I guess you didn't read my question properly (although I maybe didn't
>> have to ask it in this thread to avoid this, but I didn't want to make
>> a new thread because of David's post several days ago). I'm not trying
>> to resize anything. I just want the -999 or NaN (I make them with !
>> Values.F_NAN option btw.) values to have a value in the b array that
>> is calculated by interpolation (for instance: the IDW of the
>> surroundig pixels in a 3x3 window).
>
>> So sorry for the mixup and thanks if you guys still feel like finding

```

>> a solution for this.

>

> Hi,

> ah, now you give more information... a 3*3 window, and IDW... well, the

> "do it yourself" solution is the easiest... You can put that in a

> function or else.

>

Hi,

I am not sure if I am following this new twist (isn't IDW a type of root beer?) I should point out that IDL's built in GRIDDATA routine will accept polygonal breaks in interpolation (such as a polygon describing the break between land and water). I have found that it works well even if it is a pain to develop the polygons. That might be just what you are looking for. It is worth getting over the hump to learn the ins-and-outs of GRIDDATA if you will be doing this kind of thing often.

Cheers,

Ben
