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