Subject: Re: INTERPOLATE function - Question
Posted by wlandsman on Wed, 09 Mar 2016 19:52:40 GMT
View Forum Message <> Reply to Message

On Wednesday, March 9, 2016 at 7:34:04 AM UTC-5, dmfl...@gmail.com wrote:
> Hi all
>
> I wrote the following code because I'm interested to understand how the INTERPOLATE (bilinear) function works.
>
```
> Big = randomu(2,136,136)
> nint = size(Big, /dimensions)
>
> Small = fltarr(4,4)
> Small[0,0]=0.1
> n = size(Small, /dimensions)
> n = n[1:*]
>
> X = (n[0]-1)*findgen(nint[0])/(nint[0]-1E)
> Y = (n[0]-1)*findgen(nint[0])/(nint[0]-1E)
>
> Small_int = fltarr(nint[0],nint[1])
> Small_int = INTERPOLATE(reform(Small[*,*]), X, Y, /GRID)
```
>
> The Small array which is the array I want to interpolate has only one non-zero entry. When I interpolated from [4,4] to [136,136] I noticed that Small_int[0:44,0:44] its the non-zero part of the matrix (2025 non-zero pixels), i.e. that part of matrix affected by interpolation.
>

Your formulae for X and Y are incorrect.   The output values currently go from 0 to 3 whereas you want them to go from 0 to 4.   (In one dimension "0" refers to the left edge of the first pixel, and "4" refers to the right edge of the last pixel.)

```
Y = n[0]*findgen(nint[0])/nint[0]
```

if you do this you will the find same percentage of non-zero pixels in the small array as the big one.   In your case, 1 out of 4 pixels should be nonzero so 136/4. =34 pixels in each dimension, small_int[0:33,0:33]