

---

Subject: POLY\_2D proken. Film at 11.

Posted by [Craig DeForest](#) on Fri, 21 Aug 1998 07:00:00 GMT

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

---

I found a rather interesting bug in poly\_2d, the IDL built-in to do scaling of image data. The bilinear and spline interpolation features are designed inconsistently with the sampling feature. The bug is both in 4.x and 5.x versions of IDL.

Sampling works correctly: when scaling an original image by an integer factor, each pixel is scaled an integer number of times. But bilinear and cubic interpolation do not work the same way -- there is a 1/2-pixel offset in the output compared to linear sampling. Apparently, the interpolation algorithms wrongly regard each (old) pixel's value as resident at the \*corner\* of the (old) pixel, and not at the \*center\* of the (old) pixel.

Here's some example code:

-----  
pro break\_poly\_2d

; Generate a symmetrical image of a crosshairs

a = bytarr(9,9)

a(4,\*) = 255

a(\*,4) = 255

window,0,xsiz=9,ysiz=9

tv,a

; Scale it up by a factor of 10 using the sampling algorithm

; The output looks nice so far...

b = poly\_2d(a,[0,0.1,0,0],[0,0.1,0],0,90,90)

window,1,xsiz=90,ysiz=90

tv,b

; Scale it up by a factor of 10 using the bilinear interpolation

; algorithm. Shudder at the lack of consistency.

c = poly\_2d(a,[0,0.1,0,0],[0,0.1,0],1,90,90)

window,2,xsiz=90,ysiz=90

tv,c

; Scale it up by a factor of 10 using the bilinear interpolation

; algorithm, but offset to account for the pixel-corner bug.

; Recoil in horror at the sloppy treatment of the boundary condition.

d = poly\_2d(a,[-0.5,0.1,0,0],[-0.5,0.1,0],1,90,90)

window,3,xsiz=90,ysiz=90

tv,d

; Scale it up by a factor of 10 using the cubic spline.

```
; Laugh that at least it's broken consistently with the
; bilinear case.
e = poly_2d(a,[-0.5,0.1,0,0],[-0.5,0,0.1,0],2,90,90)
window,4,xsiz=90,ysiz=90
tv,d

end
-----
```

The best one can do is to say something inane like:

```
P1=P
P1(0) = P1(0)-0.5*keyword_set(method)
Q1=Q
Q1(0) = Q1(0)-0.5(keyword_set(method)
out = poly_2d(in,P1,Q1,method,xsize,ysize)
```

instead of

```
out = poly_2d(in,P,Q,method,xsize,ysize)
```

but even then you get wacky results near the lower and left hand boundaries of <out>.

--

I work for Stanford, \*NOT\* the government. My opinions are my own.

If you're a robot, please reply to the address in the header.

If you're human, try " zowie (at) urania . nascom . nasa . gov "