Subject: Simple? problem
Posted by Ivan Valtchanov on Tue, 09 Apr 2002 15:49:17 GMT
View Forum Message <> Reply to Message

Hi,

I have a small problem concerning a good programming techniques, so here it is:

```
---------------------
pro test,nx,ny,image

; Make some random X and Y arrays
x = randomu(s,1000)
y = randomu(s,1000)

; now give arbitrary weights for each point
w = randomu(s,1000)/100.0
; take the square
w2=2.0*w*w

; I want to construct a 2-D image with a specified dimension
image = fltarr(nx,ny)

; I want to sum up the contribution of each point as Gaussian
; with width=w to the image pixels

for i=0, nx-1 do begin
 xgi = i/float(nx)
 dx = x-xgi
 for j=0, ny-1 do begin
  ygi = j/float(ny)
  dy = y-ygi
  dr2 = dx*dx+dy*dy
  arg = -dr2/w2 > (-20.0) ; to avoid overflows
  image[i,j] =  total(exp(arg))
 endfor
endfor

return
end
--------------------------
```

This is obviously quite unoptimised - two cycles etc. Do you have any ideas, references or do you know if it is already solved in IDL?

I have looked for something similar in David Fanning pages and IDL astronomical libraries but I couldn't find something to adapt, maybe I have missed it?

Thanks.

Ivan V.

Subject: Re: Simple? problem
Posted by Jaco van Gorkom on Thu, 11 Apr 2002 09:55:38 GMT
View Forum Message <> Reply to Message

"Ivan Valtchanov" <ivanv@discovery.saclay.cea.fr> wrote in message
news:20020410101353.334cb503.ivanv@discovery.saclay.cea.fr.. .
> To describe what is my objective: this is an adaptive kernel
> smoothing, where the width of the kernel is locally variable.
> I would like to apply it to scattered points and not (for the
> moment) to images.
>
> Hope this helps?

Ok, I see now why the code is like that. Nothing really smart comes to mind, apart from vectorizing the inner loop as in my previous post. Of course, if the number of scattered points is generally smaller than the number of rows/points in the image then a loop over the points, adding their gaussian contribution to the whole image, might be more efficient.

Cheers,
jaco
