
Subject: avoiding for loops

Posted by [lbryanNOSPAM](#) on Thu, 12 Nov 1998 08:00:00 GMT

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

I've had such good luck recently with y'all that I thought I'd send another one. Although this is a little less esoteric. I need to apply a median filter all the way to the edges of an image. So I came up with putting an edge around it the width of the filter then selecting out the original region. Sort of what convol does automatically.

The following code is what I came up with. It works, but it is clunky as heck. I know some of you guys are wizzes at getting rid of loops. Care to lend a hand?

p.s. the data I'm working on array 4000x512 arrays not 100x100!

Thanks!!!

```
n = 3
bottshade = shift(dist(100),10)+randomu(100,100,100)*5.
azrange = 100
shotrange = 100
temp = fltarr(shotrange +2*n,azrange + 2*n)
temp(n,n) = bottshade
for i = 0,shotrange -1 +2*n do begin
    temp(i,0:n-1) = temp(i,n)
    temp(i,azrange-1+n:azrange-1+2*n) = temp(i,azrange +n - 2)
endfor
for i = 0,azrange -1 +2*n do begin
    temp(0:n-1,i) = temp(n,i)
    temp(shotrange-1+n:shotrange-1+2*n,i) = temp(shotrange +n - 2,i)
endfor
temp2 = median(temp,n)
bottshade = temp(n:n+shotrange-1,n:n+azrange)

end
```

Lisa
Arete Associates
Tucson, Arizona
lbryan@arete-az.com

Subject: Re: Avoiding For Loops

Posted by [Dick Jackson](#) on Mon, 01 Aug 2005 21:35:40 GMT

"sudipta" <ssarkar@forestone.com> wrote in message
news:1122923758.653063.102140@f14g2000cwb.googlegroups.com.. .

> Hello Everybody,
> My IDL problem is simple but is very crucial.
. [...]
> So i was wondering if there was a way
> to avoid the for loops and do the RMS computation in one single step?

Hi Sudipta,

Your guess that IDL should be able to do this is correct. Read on! I did a few iterations of your calculations, printing out the result, then did the whole works after that. You can save this next section in a .pro file, compile it and run it. Hope this helps.

=====

```
;; Make some sample data
```

```
data = RandomU(seed,10,14014,10,6)  
lfData = RandomU(seed,6)
```

```
Print, 'Original Way:'  
Print
```

```
;; I have reordered the loops with slowest-changing dimension in  
;; the outer loop... in IDL, this is the last dimension.
```

```
numbands = 6
```

```
for k = 0, 1 do begin ;number of fractions (10)  
  for j = 0, 1 do begin ;number of spectral combinations (14013)  
    for i = 0,1 do begin ;number of AI (10)  
      tdata = reform(data(i,j,k,*),numbands,1)  
      Print, total((tdata - lfdata)^2)  
    end  
    Print  
  end  
  Print  
end
```

```
Print, 'Un-loopy way :-)' :'  
Print
```

```
t0 = SysTime(1)
```

```

;; Extend the 6 values across all other dimensions
IfData = Reform(IfData,1,1,1,6)
IfDataExt = Rebin(IfData,10,14014,10,6)

;; Calculate RMS in one step
rms = Total((data-IfDataExt)^2, 4)

timeTaken = SysTime(1)-t0
Print, 'Time taken:'+StrTrim(timeTaken, 2)+' seconds.'
Print

;; Print 2x2x2 section for comparison
Print, rms[0:1,0:1,0:1]

END

```

=====

Printout from above program:

Original Way:

0.358912
1.03072

0.164605
0.782057

0.910758
0.800281

0.591216
1.48899

Un-loopy way :-)

Time taken:0.29700017 seconds.

0.358912 1.03072
0.164605 0.782057

0.910758 0.800281
0.591216 1.48899

=====

Cheers,

--

-Dick

Dick Jackson / dick@d-jackson.com

D-Jackson Software Consulting / <http://www.d-jackson.com>

Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392
