Subject: Re: Technique to find maximum in 100x100 element moving box
Posted by Lajos Foldy on Thu, 13 Oct 2016 18:56:46 GMT
View Forum Message <> Reply to Message

On Thursday, October 13, 2016 at 7:58:50 PM UTC+2, fawltyl...@gmail.com wrote:
> On Thursday, October 13, 2016 at 7:01:56 PM UTC+2, fawltyl...@gmail.com wrote:
>
>> Try this:
>
> Revised version, faster and without ifs:
>
> pro test
> nx=3200
> ny=3248
> m=100
>
> m2=m/2
> seed=123
> data=randomu(seed,nx,ny)
>
> tic
> tmp1=transpose(data)
> tmp2=fltarr(ny,nx,/nozero)
> FOR i = 0, nx-1 DO BEGIN
> FOR j = 0, m2 DO tmp2[j,i] = max(tmp1[0 :j+m2, i])
> FOR j = m2+1, ny-m2-1 DO tmp2[j,i] = max(tmp1[j-m2:j+m2, i])
> FOR j = ny-m2, ny-1 DO tmp2[j,i] = max(tmp1[j-m2:ny-1, i])
> ENDFOR
> tmp2=transpose(tmp2)
> data_max=fltarr(nx,ny,/nozero)
> FOR j = 0, ny-1 DO BEGIN
> FOR i = 0, m2 DO data_max[i,j] = max(tmp2[0 :i+m2, j])
> FOR i = m2+1, nx-m2-1 DO data_max[i,j] = max(tmp2[i-m2:i+m2, j])
> FOR i = nx-m2, nx-1 DO data_max[i,j] = max(tmp2[i-m2:nx-1, j])
> ENDFOR
> toc
>
> end
>
>
> regards,
> Lajos
>
> ps: the i-50:i+50 subscript range has 101 elements, not 100.

Last version, with a real sliding window. This one is about 30x faster than the original code for random data.

```
pro test
nx=3200
ny=3248
m=100
m2=m/2
seed=123
data=randomu(seed,nx,ny)

tic
tmp1=transpose(data)
tmp2=fltarr(ny,nx,/nozero)
FOR i = 0, nx-1 DO BEGIN
      FOR j = 0, m2        DO tmp2[j,i] = max(tmp1[0     :j+m2, i])
      maxi=tmp2[m2,i]
      FOR j = m2+1, ny-m2-1 DO begin
         if maxi eq tmp1[j-m2-1, i] then begin
            maxi=max(tmp1[j-m2:j+m2, i])
         endif else maxi=maxi>tmp1[j+m2, i]
         tmp2[j,i]=maxi
      endfor
      FOR j = ny-m2, ny-1   DO tmp2[j,i] = max(tmp1[j-m2  :ny-1, i])
   ENDFOR
tmp2=transpose(tmp2)
data_max=fltarr(nx,ny,/nozero)
FOR j = 0, ny-1 DO BEGIN
      FOR i = 0, m2        DO data_max[i,j] = max(tmp2[0     :i+m2, j])
      maxi=data_max[m2,j]
      FOR i = m2+1, nx-m2-1 DO begin
         if maxi eq tmp2[i-m2-1, j] then begin
            maxi=max(tmp2[i-m2:i+m2, j])
         endif else maxi=maxi>tmp2[i+m2, j]
         data_max[i,j]=maxi
      endfor
      FOR i = nx-m2, nx-1   DO data_max[i,j] = max(tmp2[i-m2  :nx-1, j])
   ENDFOR
toc
end
```

regards,
Lajos

---