Subject: Re: TRIANGULATE. Finding contiguous cells efficiently?
Posted by Wox on Mon, 26 Feb 2007 12:27:49 GMT
View Forum Message <> Reply to Message

On 25 Feb 2007 19:37:54 -0800, "Libertan" <tbethell@umich.edu> wrote:

> In case it has gone into the abyss

It has :-)

> thoughts:
> 1) The connectivity list might be fruitful after all.

I don't think this could make things faster.

> 2) Also I think that in these looped codes, the accumulated info in
> the loop can be used to increasingly reduce the remaining workload.

Well ... one could think of things like: "deleting triangles which are already used 3 times". Example below gives an 18% improvement on my PC, when using the shrinking-triangles.

```
pro test
x=RANDOMU(seed,10000,/normal)*10
y=RANDOMU(seed,10000,/normal)*10
nnodes=n_elements(x)
TRIANGULATE, X, Y, triangles
s=size(triangles,/dimensions)

; 1. With loop
Tm=systime(1)
ctriangles=fltarr(s[0],s[1],3)
for i=0,s[1]-1 do begin
 t=total((triangles eq triangles[0,i])+$
 (triangles eq triangles[1,i])+$
 (triangles eq triangles[2,i]),1)

 ind=where(t eq 2,ct)
 if ct ne 0 then ctriangles[*,i,0:ct-1]=triangles[*,ind]
endfor
; Third dimension gives the three contiguous neighbours
print,systime(1)-Tm
Tm=systime(1)

; 2. With loop + shrinking triangles
Tm=systime(1)
```

```
ctriangles=fltarr(s[0],s[1],3)
trshrink=triangles
nused=bytarr(s[1])
indtr=lindgen(s[1])
for i=0,s[1]-1 do begin
 t=total((trshrink eq triangles[0,i])+$
 (trshrink eq triangles[1,i])+$
 (trshrink eq triangles[2,i]),1)

 ind=where(t eq 2,ct)
 if ct ne 0 then begin
  ctriangles[*,i,0:ct-1]=trshrink[*,ind]
  nused[ind]++

  tmp=where(nused eq
3,ct,COMPLEMENT=ind,NCOMPLEMENT=ct2)
  if ct ne 0 then begin
   if ct2 ne 0 then begin
    nused=nused[ind]
    trshrink=trshrink[*,ind]
   endif
  endif
 endif
endfor
; Third dimension gives the three contiguous neighbours
print,systime(1)-Tm
Tm=systime(1)
end
```