Subject: Re: Nearest Neighbor ... again!
Posted by Fabzi on Mon, 12 Jul 2010 08:14:09 GMT
View Forum Message <> Reply to Message

> These two methods give a different result. Try running the code below.
> The red and green lines connect the nearest neighbours from method 1
> and method 2. You should see only red lines, but you see some green
> lines too...

Yes, the first method is actually the fastest (especially when working
on
huge datasets).

>
> This doesn't answer your question about finding the 4 closest
> neighbours however. Couldn't you use griddata with bilinear
> interpolation and get the "4 corners" from the interpolated value?
> I'll think about it :-).

Thanks!

>
> The real question is, what are you going to do when you have the 4
> nearest neighbours? What are you trying to achieve that can't be done
> with IDL's gridding and interpolation routines?

Yes, I probably want to do something that IDL might solve better than
me.
I want to fit a dataset with quality assessment (MODIS) on an other
grid,
and test all four neirest points to take the best quality value. This
concerns
only few grid points, as most of time the points are either "all good"
or
"all bad".

It was also interesting in itself, to be able to have the four neirest
in hand !

I see that my request has something "unusual", and will probably not
be solved
without long computing times. I will try to address my problem
differently.

Thanks again, I learned a lot already just trying to do it.

>
> pro ConnectGrids

```
>
>  ; First grid
>  n1 = 100
>  seed = -121147L
>  x1 = round(RANDOMU(seed, n1)*n1)
>  y1 = round(RANDOMU(seed, n1)*n1)
>
>  ; Second grid
>  n2 = 100
>  x2 = round(RANDOMU(seed, n2)*n2)
>  y2 = round(RANDOMU(seed, n2)*n2)
>
>  ; Find closest: method 1
>  triangulate, x1, y1, c ; Compute Delaunay triangulation
>  iconnect = GRIDDATA(x1,y1, LINDGEN(n1), Xout=x2, Yout=y2,$
>         /NEAREST_N,TRIANGLES =c)
>
>  ; Find closest: method 2
>  iconnect2=iconnect
>  for i=0l,n2-1 do begin
>         tmp=min((x2[i]-x1)^2.+(y2[i]-y1)^2.,ind)
>         iconnect2[i]=ind[0]
>  endfor
>
>  ; Plot result
>  device,decompose=0
>  loadct,39
>  window
>  n=n1>n2
>  plot,x1,y1,psym=1,xrange=[-10,n+9],yrange=[-10,n+9],/xs,/ys
>  oplot,x2,y2,psym=1,color=100
>  for i=0,n2-1 do
>   plots,[x1[iconnect[i]],x2[i]],[y1[iconnect[i]],y2[i]],color= 150
>  for i=0,n2-1 do plots,$
>         [x1[iconnect2[i]],x2[i]],[y1[iconnect2[i]],y2[i]],color=250
>
>  end
```