
Subject: Re: Inside rectangle

Posted by [R.Bauer](#) on Tue, 02 Jul 2002 17:40:04 GMT

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

Neil Talsania wrote:

>
> I am relatively new to IDL , and am wondering how to do the following. I
> want to be able to have user input 4 corner coordiates, then check to see if
> a particular point is inside the rectangle. That is pretty easy, i think.
> But then I want to be able to add additional sets of coordinates to check. I
> want this to be unlimited. Basically allowing the user to add rectangles to
> check until he is ready to stop.
>
> Now I think I want to make a structure to hold the 4 corner coordinates.
> Then I need to create an array of these structures, but I want that array to
> be able to grow. Then I will loop through that array (For loop) calling a
> function that will check to see if the point is inside the coordinates.
>
> So the crux of my problem (I think) is that I need the array size to
> increase every time the user adds a set. But I dont know ho to do this.
>
> Any clues would be greatly appreciated.
>
> Neil

Dear Neil

first define a named structure like:

```
struct=create_struct(name='rectangle','coord',make_array(4,/ float))
```

the grown of the structure is done by concatenating e.g.

```
for i=0,5 do $  
if size(result,/n_elements) eq 0 then result=struct else  
result=[result,struct]
```

```
help,result  
STRUCT      STRUCT  = -> RECTANGLE Array[6]
```

regards
Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)

Forschungszentrum Juelich

email: R.Bauer@fz-juelich.de

a IDL library at Forschungszentrum Juelich

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html
=====

Subject: Re: Inside rectangle

Posted by [btupper](#) on Tue, 02 Jul 2002 18:58:04 GMT

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

On Tue, 2 Jul 2002 12:53:03 -0400, "Neil Talsania"

<neil.talsania@kodak.com> wrote:

> I am relatively new to IDL , and am wondering how to do the following. I
> want to be able to have user input 4 corner coordinates, then check to see if
> a particular point is inside the rectangle. That is pretty easy, i think.
> But then I want to be able to add additional sets of coordinates to check. I
> want this to be unlimited. Basically allowing the user to add rectangles to
> check until he is ready to stop.

Hello,

This came up a while ago in regards to Delauney triangulation meshes... but what's a triangle but a pointy headed rectangle. The following might help. You should note that you will need David Fanning's LoadData function and Liam Gumley's ImDisp procedure. Also, you might want to rename the object... I wasn't very original when I named it.

Once you have all the code do the following from the command line...

```
IDL> .compile anroigroup__define
```

```
IDL> example
```

I hope this helps.

Ben

```
;-----START
```

```
;-----
```

```
; WhichROI
```

```
;-----
```

```
FUNCTION AnROIGroup::WhichROI, X, Y, Z, $
```

```
count = count, type = type, index = index
```

```
;This function returns the IDLanROI (OR the ROIs)  
;that contain the point X,Y,Z Only one point is tested  
;per call. If more than ROI contains the point, then  
;an array of IDLanROI objects is returned  
;  
;X,Y,Z see IDLanROI::ContainsPoints  
; ONLY a single point checked  
;COUNT the number of ROIs found that contain X, Y, Z  
;TYPE an array of the values returned by the  
; IDLanROI::ContainsPoints method  
;INDEX the postional index of the IDLanROI  
; within the AnROIGroup
```

```
np = n_params()  
Case np of  
  1: data = x[0:2]  
  2: data = [x[0], y[0]]  
  3: data = [X[0],Y[0], Z[0]]  
EndCase
```

```
For i = 0L, self->Count() -1 DO Begin
```

```
ROI = Self->Get(Position = i)  
r = ROI->ContainsPoints(data)
```

```
If r GT 0 Then Begin  
  If n_elements(Arr) EQ 0 Then Begin  
    Arr = ROI  
    type = r  
    Index = i  
  EndIf Else Begin  
    Arr = [Arr,ROI]  
    type = [type,r]  
    Index = [index,i]  
  EndElse  
EndIf
```

```
EndFor
```

```
Count = n_elements(Arr)  
Return,Arr  
END ;WhichROI
```

```
;-----  
; Definition
```

```
;-----  
PRO AnROIGroup__Define
```

```
struct = {AnROIGroup, $  
  
  Inherits IDLanROIGroup}  
  
END ;AnROIGroup
```

```
;-----  
; EXAMPLE  
;-----
```

```
PRO Example
```

```
bottom = 32  
loadCT, 0, bottom = bottom  
Tek_Color
```

```
XYZ = LoadData(14)
```

```
Triangulate, XYZ[0,*],XYZ[1,*], tri
```

```
Surf = TriGrid( XYZ[0,*],XYZ[1,*],XYZ[2,*], tri,$  
  xgrid = xg, ygrid = yg)
```

```
xRange = [Min(xg), Max(xg)]  
yRange = [Min(yg), Max(yg)]
```

```
ImDisp, Surf, /axis,/erase, xrange = xrange, $  
  yrange = yrange, bottom = bottom, $  
  Color = 1, Background = 0
```

```
oPlot, XYZ[0,*],XYZ[1,*], psym = 6, color = 2
```

```
Group = OBJ_NEW('AnROIGroup')
```

```
For i = 0, n_elements(Tri)/3 -1 Do Begin  
  index = [tri[*],i], tri[0,i]]  
  oPlot, XYZ[0,index],XYZ[1,index], color = 3  
  o = OBJ_NEW('IDLanROI', XYZ[0,index],XYZ[1,index])  
  Group->Add, o  
EndFor
```

```
myX = [-110.0]  
myY = [29.0]
```

```

PlotS, myX, myY, psym = 2, color = 4

Arr = Group->WhichROI(myX, myY, $
count = count, type = type, index = index)

If Count GT 0 Then Begin
  Help, arr
  print, count
  print, type
  print, index

  Arr[0] ->GetProperty, Data = Data
  oPlot, Data[0,*], Data[1,*], thick = 2, color = 5
EndIf

Obj_Destroy, Group
END
;-----END

```

Subject: Re: Inside rectangle
 Posted by [Rick Towler](#) on Tue, 02 Jul 2002 21:49:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Neil Talsania" <neil.talsania@kodak.com> wrote > Now I think I want to make a structure to hold the 4 corner coordinates.

- > Then I need to create an array of these structures, but I want that array to
- > be able to grow. Then I will loop through that array (For loop) calling a
- > function that will check to see if the point is inside the coordinates.
- >
- > So the crux of my problem (I think) is that I need the array size to
- > increase every time the user adds a set. But I dont know ho to do this.

Reimar's suggestion will work but if you are carting this data around your application in a structure, at some point you'll want to use a pointer to reference your array of structures.

But better yet, why not use a linked list? Then you can easily add and remove individual rectangles and not have to deal with the details. You can download David Fanning's linkedlist__define.pro from his website (www.dfanning.com). Since the linked list is an object it is simple to cart around in a structure. His object returns a pointer to the data contained in a node and you will need to be aware of precedence when dereferencing the pointer:

imagine 'll' is my linked list object with all of my rectangle structures I

can get a pointer 'p' to the first structure in the list:

```
p = ll->get_item(0)
```

To get the data that the pointer refers to I need to use parentheses to ensure that IDL dereferences the pointer p before trying to access the structure elements:

```
IDL> print,(*p).coord  
    0.000000  0.000000  0.000000  0.000000
```

-Rick

Subject: Re: Inside rectangle

Posted by [Pavel A. Romashkin](#) on Wed, 03 Jul 2002 17:28:37 GMT

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

Using an array of structures seems cumbersome to me. If you choose to follow this path, why not simply use a FLTARR(4, x) of corner coordinates? An array is easily grown, and if you stick with rectangles, you always have 4 points to put into the array. Then, you can loop through it as much as you'd like.

It looks to me also that using rectangles is just a case of a general ROI. IDLanROI object handles all the required data processing. All you have to do is keep adding coordinates to it. And you don't have to know IDL objects to use it, it is very straightforward. It is NOT object graphics.

Cheers,
Pavel

Neil Talsania wrote:

>

> I am relatively new to IDL , and am wondering how to do the following. I
> want to be able to have user input 4 corner coordiates, then check to see if
> a particular point is inside the rectangle. That is pretty easy, i think.
> But then I want to be able to add additional sets of coordinates to check. I
> want this to be unlimited. Basically allowing the user to add rectangles to
> check until he is ready to stop.

>

> Now I think I want to make a structure to hold the 4 corner coordinates.
> Then I need to create an array of these structures, but I want that array to
> be able to grow. Then I will loop through that array (For loop) calling a
> function that will check to see if the point is inside the coordinates.

>

> So the crux of my problem (I think) is that I need the array size to
> increase every time the user adds a set. But I dont know ho to do this.

>

> Any clues would be greatly appreciated.
>
> Neil
