
Subject: Re: Polygon Clipping Algo in IDL
Posted by [raval.chintan](#) on Tue, 13 Dec 2005 05:29:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Mark,

Thank You.

Regards,
Chintan Raval.

Subject: Re: Polygon Clipping Algo in IDL
Posted by [raval.chintan](#) on Tue, 13 Dec 2005 06:32:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Mark,

How can i give two polygon as an input to your program.

Suppose i have polygon 1) `[[0,0],[0,3],[3,3],[3,0]]`
2) `[[1,1],[1,4],[4,4],[4,1]]`

Here my polygons are rectangle. First polygon contains upper left point as `[0,0]` and lower right as `[3,3]`.
and second polygon contains upper left as `[1,1]` and lower right as `[4,4]` .

Now my result (output) polygon will contain `[1,1]` as upper left and `[3,3]` as lower right .

Regards
Chintan

Subject: Re: Polygon Clipping Algo in IDL
Posted by [Mark Hadfield](#) on Tue, 13 Dec 2005 23:14:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

raval.chintan@gmail.com wrote:

> Dear Mark,
>
> How can i give two polygon as an input to your program.
> Suppose i have polygon 1) `[[0,0],[0,3],[3,3],[3,0]]`
> 2) `[[1,1],[1,4],[4,4],[4,1]]`
>
> Here my polygons are rectangle. First polygon contains upper left
> point as `[0,0]` and lower right as `[3,3]`.

> and second polygon contains upper left as [1,1] and lower right as
> [4,4] .
>
> Now my result (output) polygon will contain [1,1] as upper left and
> [3,3] as lower right .
>
> Regards
> Chintan
>

Ok. Let's (arbitrarily) consider the first polygon as the one to be clipped (the clippee) and the second as the one to clip to (the clipper). MGH_POLYCLIP clips a polygon to a line, so we will need to apply it 4 times. It turns out that I have a function that does this, called MGH_POLYBOX, attached. It's not in the Motley library but it probably should be.

The calling sequence is

```
result = MGH_POLYBOX(xclip, yclip, polin)
```

where xclip is a 2-element vector specifying the clipping values in the X direction, yclip is a 2-element vector specifying the clipping values in the Y direction and polin is a [2,n] vector defining the polygon to be clipped.

So in your case

```
IDL> xclip = [1,4]
IDL> yclip = [1,4]
IDL> polin = [[0,0],[0,3],[3,3],[3,0]]
IDL> print, mgh_polybox(xclip, yclip, polin)
      1      1
      1      3
      3      3
      3      1
```

--

Mark Hadfield "Kei puwaha te tai nei, Hoesa tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

```
;+
; NAME:
; MGH_POLYBOX
;
; PURPOSE:
```

```

; Clip an arbitrary polygon on the X-Y plane to a box (a rectangle
; defined by X and Y limits) using the Sutherland-Hodgman algorithm.
;
; CATEGORY:
; Graphics, Region of Interest, Geometry
;
; CALLING SEQUENCE:
; result = MGH_POLYBOX(xclip, yclip, polin, COUNT=count)
;
; RETURN VALUE
; The function returns [2,n] vector defining the clipped polygon. The
; second dimension will equal the value of the COUNT argument, except
; where this is 0 in which the return value is -1.
;
; ARGUMENTS
; xclip  A 2-element vector specifying the clipping values in the
;        X direction
;
; yclip  A 2-element vector specifying the clipping values in the
;        Y direction
;
; polin  A [2,n] vector defining the polygon to be clipped.
;
; KEYWORDS
; COUNT  Associate this keyword with a named variable to return
;        the number of vertices in the clipped polygon.
;
; PROCEDURE:
; The polygon is clipped to each edge in turn using the Sutherland-Hodgman
; algorithm.
;
; This function is based on JD Smith's POLYCLIP function. He can take all
; of the credit and none of the blame.
;
; MODIFICATION HISTORY:
; Mark Hadfield, 2001-10:
;   I wrote thsi first as a stand-alone function, based on JD Smith's
;   POLYCLIP, then modified it so that it just calls MGH_POLYCLIP
;   up to 4 times.
;-

```

```
function mgh_polybox, xc, yc, polin, COUNT=count
```

```

compile_opt DEFINT32
compile_opt STRICTARR

```

```
polout = mgh_polyclip(xc[0], 0B, 0B, polin, COUNT=count)
```

if count eq 0 then return, polout

polout = mgh_polyclip(xc[1], 0B, 1B, polout, COUNT=count)

if count eq 0 then return, polout

polout = mgh_polyclip(yc[0], 1B, 0B, polout, COUNT=count)

if count eq 0 then return, polout

polout = mgh_polyclip(yc[1], 1B, 1B, polout, COUNT=count)

return, polout

end

File Attachments

1) [mgh_polybox.pro](#), downloaded 152 times
