
Subject: polygon intersection routine in IDL
Posted by [Weihua FANG](#) on Wed, 14 Feb 2007 03:15:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

hi, all,

I need a routine to get the exact intersection, to be returned as a polygon, of 2 polygons (could be convex or concave) . I did some search and found similar routines from the libs of motley and Dr. D.Faning. but seems not the exact one i am looking for.

Is there anybody knows such a routine in IDL?

thanks a lot.

Weihua

Subject: Re: polygon intersection routine in IDL
Posted by [Weihua FANG](#) on Wed, 14 Feb 2007 11:36:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

It works fine now.

Thank you

> In article <1171433274.480043.121...@q2g2000cwa.googlegroups.com>, "Weihua FANG"
<weihua.f...@gmail.com> writes:

>

>

>

>> Hi, dear mati,

>

>> I tested your function by 2 samples. One returned the overlap polygon
>> successfully. but the second one failed. is there anything wrong in my
>> code?

>

>> thank you in advance.

>

>> pro test_polygon_overlap

>

>> shape1 =3D fltarr(2,4)

>> shape2 =3D fltarr(2,4)

>> shape1[0,0] =3D 0.1

>> shape1[0,1] =3D 1.1

>> shape1[0,2] =3D 1.2

>> shape1[0,3] =3D 0.2

```

>> shape1[1,0] =3D 0.2
>> shape1[1,1] =3D 0.15
>> shape1[1,2] =3D 1.1
>> shape1[1,3] =3D 1.2
>
>> shape2[0,0] =3D 0.5
>> shape2[0,1] =3D 1.5
>> shape2[0,2] =3D 1.6
>> shape2[0,3] =3D 0.4
>> shape2[1,0] =3D 0.5
>> shape2[1,1] =3D 0.6
>> shape2[1,2] =3D 1.7
>> shape2[1,3] =3D 1.5
>
>> shape3 =3D shape_overlap (shape1, shape2, exists =3D exs)
>
>> print, exs
>
>> shape1 =3D fltarr(2,17)
>> shape1 [*,0] =3D [116.44991,41.441910]
>> shape1 [*,1] =3D [116.57357,41.374821]
>> shape1 [*,2] =3D [116.64723,41.312994]
>> shape1 [*,3] =3D [116.57488,41.002544]
>> shape1 [*,4] =3D [116.36967,40.928878]
>> shape1 [*,5] =3D [116.16840,40.984128]
>> shape1 [*,6] =3D [116.21576,41.053847]
>> shape1 [*,7] =3D [116.06185,41.099889]
>> shape1 [*,8] =3D [116.10263,41.168293]
>> shape1 [*,9] =3D [116.13683,41.214334]
>> shape1 [*,10] =3D [116.16577,41.264322]
>> shape1 [*,11] =3D [116.18156,41.319572]
>> shape1 [*,12] =3D [116.22891,41.344565]
>> shape1 [*,13] =3D [116.31442,41.357720]
>> shape1 [*,14] =3D [116.29337,41.387976]
>> shape1 [*,15] =3D [116.34468,41.422178]
>> shape1 [*,16] =3D [116.44991,41.441910]
>
>> shape2 =3D fltarr(2,6)
>> shape2 [*,0] =3D [116.21839,41.528731]
>> shape2 [*,1] =3D [116.61698,41.524784]
>> shape2 [*,2] =3D [116.72748,41.141984]
>> shape2 [*,3] =3D [116.30916,41.203810]
>> shape2 [*,4] =3D [116.13552,41.289316]
>> shape2 [*,5] =3D [116.21839,41.528731]
>
>> xrange =3D [116.06185 ,116.72748]
>> yrange =3D [40.928879 ,41.528732]
>

```

```

>> plot, shape1[0,*], shape1[1,*], xrange =3D xrange , yrange =3D yrange
>> oplot, shape2[0,*], shape2[1,*]
>
>> shape3=3D shape_overlap (shape1, shape2, exists =3D exs)
>
>> print, exs
>
>> end
>
> Well, thank you. I didn't use this routine for nearly a decade,
> didn't notice that it has a bug. OK, go to line 72 of the routine,
> the one which has a statement starting with
>
>     if Shape_area....
>
> And replace it with
>
>     if Shape_area(sec) lt 0 then sec = reverse(sec,2)
>
> Should work now, let me know if it doesn't
>
> Mati Meron          | "When you argue with a fool,
> m...@cars.uchicago.edu | chances are he is doing just the same"-
>
>

```

Subject: Re: polygon intersection routine in IDL
 Posted by [mmeron](#) on Wed, 14 Feb 2007 18:07:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1171452966.005192.305170@a34g2000cwb.googlegroups.com>, "Weihua FANG"
 <weihua.fang@gmail.com> writes:

> It works fine now.

>

> Thank you

>

You're welcome. Take a look at the other routines with names starting
 with SHAPE_ , they all deal with polygons.

>

>> In article <1171433274.480043.121...@q2g2000cwa.googlegroups.com>, "Weihu=
 > a FANG" <weihua.f...@gmail.com> writes:

>>

>>

>>

>>> Hi, dear mati,

```

>>
>>> I tested your function by 2 samples. One returned the overlap polygon
>>> successfully. but the second one failed. is there anything wrong in my
>>> code?
>>
>>> thank you in advance.
>>
>>> pro test_polygon_overlap
>>
>>> shape1 =3D3D fltarr(2,4)
>>> shape2 =3D3D fltarr(2,4)
>>> shape1[0,0] =3D3D 0.1
>>> shape1[0,1] =3D3D 1.1
>>> shape1[0,2] =3D3D 1.2
>>> shape1[0,3] =3D3D 0.2
>>> shape1[1,0] =3D3D 0.2
>>> shape1[1,1] =3D3D 0.15
>>> shape1[1,2] =3D3D 1.1
>>> shape1[1,3] =3D3D 1.2
>>
>>> shape2[0,0] =3D3D 0.5
>>> shape2[0,1] =3D3D 1.5
>>> shape2[0,2] =3D3D 1.6
>>> shape2[0,3] =3D3D 0.4
>>> shape2[1,0] =3D3D 0.5
>>> shape2[1,1] =3D3D 0.6
>>> shape2[1,2] =3D3D 1.7
>>> shape2[1,3] =3D3D 1.5
>>
>>> shape3 =3D3D shape_overlap (shape1, shape2, exists =3D3D exs)
>>
>>> print, exs
>>
>>> shape1 =3D3D fltarr(2,17)
>>> shape1 [*,0] =3D3D [116.44991,41.441910]
>>> shape1 [*,1] =3D3D [116.57357,41.374821]
>>> shape1 [*,2] =3D3D [116.64723,41.312994]
>>> shape1 [*,3] =3D3D [116.57488,41.002544]
>>> shape1 [*,4] =3D3D [116.36967,40.928878]
>>> shape1 [*,5] =3D3D [116.16840,40.984128]
>>> shape1 [*,6] =3D3D [116.21576,41.053847]
>>> shape1 [*,7] =3D3D [116.06185,41.099889]
>>> shape1 [*,8] =3D3D [116.10263,41.168293]
>>> shape1 [*,9] =3D3D [116.13683,41.214334]
>>> shape1 [*,10] =3D3D [116.16577,41.264322]
>>> shape1 [*,11] =3D3D [116.18156,41.319572]
>>> shape1 [*,12] =3D3D [116.22891,41.344565]
>>> shape1 [*,13] =3D3D [116.31442,41.357720]

```

```

>>> shape1 [*,14] =3D3D [116.29337,41.387976]
>>> shape1 [*,15] =3D3D [116.34468,41.422178]
>>> shape1 [*,16] =3D3D [116.44991,41.441910]
>>
>>> shape2 =3D3D fltarr(2,6)
>>> shape2 [*,0] =3D3D [116.21839,41.528731]
>>> shape2 [*,1] =3D3D [116.61698,41.524784]
>>> shape2 [*,2] =3D3D [116.72748,41.141984]
>>> shape2 [*,3] =3D3D [116.30916,41.203810]
>>> shape2 [*,4] =3D3D [116.13552,41.289316]
>>> shape2 [*,5] =3D3D [116.21839,41.528731]
>>
>>> xrange =3D3D [116.06185 ,116.72748]
>>> yrange =3D3D [40.928879 ,41.528732]
>>
>>> plot, shape1[0,*], shape1[1,*], xrange =3D3D xrange , yrange =3D3D yran=
> ge
>>> oplot, shape2[0,*], shape2[1,*]
>>
>>> shape3=3D3D shape_overlap (shape1, shape2, exists =3D3D exs)
>>
>>> print, exs
>>
>>> end
>>
>> Well, thank you. I didn't use this routine for nearly a decade,
>> didn't notice that it has a bug. OK, go to line 72 of the routine,
>> the one which has a statement starting with
>>
>>     if Shape_area....
>>
>> And replace it with
>>
>>     if Shape_area(sec) lt 0 then sec =3D reverse(sec,2)
>>
>> Should work now, let me know if it doesn't
>>
>> Mati Meron | "When you argue with a fool,
>> m...@cars.uchicago.edu | chances are he is doing just the same"-=
> =D2=FE=B2=D8=B1=BB=D2=FD=D3=C3=CE=C4=D7=D6 -
>>
>> - =CF=D4=CA=BE=D2=FD=D3=C3=B5=C4=CE=C4=D7=D6 -
>
>

```

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: polygon intersection routine in IDL
Posted by [JD Smith](#) on Wed, 14 Feb 2007 19:05:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 13 Feb 2007 19:15:15 -0800, Weihua FANG wrote:

> hi, all,
>
> I need a routine to get the exact intersection, to be returned as a
> polygon, of 2 polygons (could be convex or concave) . I did some search
> and found similar routines from the libs of motley and Dr. D.Faning. but
> seems not the exact one i am looking for.
>
> Is there anybody knows such a routine in IDL?

I have such a routine called POLYCLIP, for clipping arbitrary polygons to a rectangular grid, and I believe Mark Hadfield reimplemented it in his library. I also have a auto-compiling C DLM version of the Sutherland Hodgeman clipper which is much (50x) faster than the internal IDL version, when you have very many polygons to clip. It's coming out with another package soon.

JD
