Subject: Re: Finding the intersection of 2 short lines
Posted by ben.bighair on Mon, 13 Feb 2012 12:48:48 GMT
View Forum Message <> Reply to Message

On Feb 13, 5:39 am, Jimmy <jimmyb...@gmail.com> wrote:
> Hi everyone.
>
> I'm trying to find a way of detecting the crossing point of two short
> lines (as part of a larger problem with detecting overlapping
> polygons). I've used intersect.pro which gives me the general
> intersect, but then I have to use the mother of all ugly if statements
> (which I'm having trouble getting working) to detect whether the
> intersect is actually on my short lines or extended away somewhere
> else.
>
> Is there a more elegant way of doing this? The code needs to find
> whether the two lines cross, and return the intersection point.
>

Hi,

I have often used Paul Bourke's geometry webpages as a good starting
point, like this one for line intersections (2d)...

http://paulbourke.net/geometry/lineline2d/

I used that stuff to develop these (you'll need David Fanning's cg*
routines)

http://dl.dropbox.com/u/8433654/pbourke_lines.zip

Cheers,
Ben

Subject: Re: Finding the intersection of 2 short lines
Posted by David Fanning on Mon, 13 Feb 2012 13:42:11 GMT
View Forum Message <> Reply to Message

ben.bighair writes:

> I have often used Paul Bourke's geometry webpages as a good starting
> point, like this one for line intersections (2d)...
>
> http://paulbourke.net/geometry/lineline2d/
>
> I used that stuff to develop these (you'll need David Fanning's cg*
> routines)

>
> http://dl.dropbox.com/u/8433654/pbourke_lines.zip

You might need to put a Forward_Function command
in the test programs (or just compile the program
code twice) to get the test programs to run.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Finding the intersection of 2 short lines
Posted by jimmybobs on Mon, 13 Feb 2012 15:15:05 GMT

View Forum Message <> Reply to Message

On Feb 13, 12:48 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
> On Feb 13, 5:39 am, Jimmy <jimmyb...@gmail.com> wrote:
>
>> Hi everyone.
>
>> I'm trying to find a way of detecting the crossing point of two short
>> lines (as part of a larger problem with detecting overlapping
>> polygons). I've used intersect.pro which gives me the general
>> intersect, but then I have to use the mother of all ugly if statements
>> (which I'm having trouble getting working) to detect whether the
>> intersect is actually on my short lines or extended away somewhere
>> else.
>
>> Is there a more elegant way of doing this? The code needs to find
>> whether the two lines cross, and return the intersection point.
>
> Hi,
>
> I have often used Paul Bourke's geometry webpages as a good starting
> point, like this one for line intersections (2d)...
>
> http://paulbourke.net/geometry/lineline2d/
>
> I used that stuff to develop these (you'll need David Fanning's cg*

> routines)
>
> http://dl.dropbox.com/u/8433654/pbourke_lines.zip
>
> Cheers,
> Ben

Hi Ben,

Thanks very much for this, works like a charm. I had looked at the website you linked, but missed the crucial line at the bottom: "The equations apply to lines, if the intersection of line segments is required then it is only necessary to test if ua and ub lie between 0 and 1. Whichever one lies within that range then the corresponding line segment contains the intersection point. If both lie within the range of 0 to 1 then the intersection point is within both line segments". Reading your code, this appears to be how you find it?

Thanks again,
Jimmy

Subject: Re: Finding the intersection of 2 short lines
Posted by jimmybobs on Mon, 13 Feb 2012 15:18:06 GMT

View Forum Message <> Reply to Message

On Feb 13, 1:42 pm, David Fanning <n...@idlcoyote.com> wrote:
> ben.bighair writes:
>> I have often used Paul Bourke's geometry webpages as a good starting
>> point, like this one for line intersections (2d)...
>
>> http://paulbourke.net/geometry/lineline2d/
>
>> I used that stuff to develop these (you'll need David Fanning's cg*
>> routines)
>
>> http://dl.dropbox.com/u/8433654/pbourke_lines.zip
>
> You might need to put a Forward_Function command
> in the test programs (or just compile the program
> code twice) to get the test programs to run.
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Thanks David - when compiling it in the IDE I did need to do it twice as you said. Funnily enough when it was tucked away in the IDL lib folder and called from my code in the working directory it seemd to work unaltered.

Jimmy

Subject: Re: Finding the intersection of 2 short lines
Posted by jimmybobs on Mon, 13 Feb 2012 15:51:31 GMT

View Forum Message <> Reply to Message

On Feb 13, 12:48 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
> On Feb 13, 5:39 am, Jimmy <jimmyb...@gmail.com> wrote:
>
>> Hi everyone.
>
>> I'm trying to find a way of detecting the crossing point of two short
>> lines (as part of a larger problem with detecting overlapping
>> polygons). I've used intersect.pro which gives me the general
>> intersect, but then I have to use the mother of all ugly if statements
>> (which I'm having trouble getting working) to detect whether the
>> intersect is actually on my short lines or extended away somewhere
>> else.
>
>> Is there a more elegant way of doing this? The code needs to find
>> whether the two lines cross, and return the intersection point.
>
> Hi,
>
> I have often used Paul Bourke's geometry webpages as a good starting
> point, like this one for line intersections (2d)...
>
> http://paulbourke.net/geometry/lineline2d/
>
> I used that stuff to develop these (you'll need David Fanning's cg*
> routines)
>
> http://dl.dropbox.com/u/8433654/pbourke_lines.zip
>
> Cheers,
> Ben

Hi,

My original reply seems to have been eaten. Your PB_LINES_INTERSECTION
routine worked like a charm, thank you.

I had read that page you linked to, but had missed the vita line at
the bottom : "The equations apply to lines, if the intersection of
line segments is required then it is only necessary to test if ua and
ub lie between 0 and 1. Whichever one lies within that range then the
corresponding line segment contains the intersection point. If both
lie within the range of 0 to 1 then the intersection point is within
both line segments. "

Thanks,

Jimmy

Subject: Re: Finding the intersection of 2 short lines
Posted by ben.bighair on Mon, 13 Feb 2012 16:50:43 GMT
View Forum Message <> Reply to Message

On Feb 13, 10:15 am, Jimmy <jimmyb...@gmail.com> wrote:
> On Feb 13, 12:48 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
>
>
>
>> On Feb 13, 5:39 am, Jimmy <jimmyb...@gmail.com> wrote:
>
>>> Hi everyone.
>
>>> I'm trying to find a way of detecting the crossing point of two short
>>> lines (as part of a larger problem with detecting overlapping
>>> polygons). I've used intersect.pro which gives me the general
>>> intersect, but then I have to use the mother of all ugly if statements
>>> (which I'm having trouble getting working) to detect whether the
>>> intersect is actually on my short lines or extended away somewhere
>>> else.
>
>>> Is there a more elegant way of doing this? The code needs to find
>>> whether the two lines cross, and return the intersection point.
>
>> Hi,
>
>> I have often used Paul Bourke's geometry webpages as a good starting
>> point, like this one for line intersections (2d)...
>
>> http://paulbourke.net/geometry/lineline2d/
>

>> I used that stuff to develop these (you'll need David Fanning's cg*
>> routines)
>
>> http://dl.dropbox.com/u/8433654/pbourke_lines.zip
>
>> Cheers,
>> Ben
>
> Hi Ben,
>
> Thanks very much for this, works like a charm. I had looked at the
> website you linked, but missed the crucial line at the bottom: "The
> equations apply to lines, if the intersection of line segments is
> required then it is only necessary to test if ua and ub lie between 0
> and 1. Whichever one lies within that range then the corresponding
> line segment contains the intersection point. If both lie within the
> range of 0 to 1 then the intersection point is within both line
> segments". Reading your code, this appears to be how you find it?
>
> Thanks again,
> Jimmy

Hi,

Yikes! I don't remember it very well, but I think that is what this
line does:

onSegment = [(ua GE ZERO) AND (ua LE ONE), (ub GE ZERO) AND (ub LE
ONE)]

Cheers,
Ben