Subject: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Tue, 03 Mar 2015 12:23:55 GMT
View Forum Message <> Reply to Message

Hi,
as already discussed in previous posts, I'm having trouble with annotations on images. I have a minimal example to show three problems.
First the example:

```
pro mouseMovetest
img = dist(500)
tlb = widget_base(/column)
wWindow = widget_window(tlb, xsize=500.0, ysize=500.0)
widget_control, tlb, /realize
widget_control, wWindow, get_value=oWin
io = image(img, image_dimensions=[500.0,500.0], current=oWin, margin=0)
nPoints = 100.0
points = (2.0 * !PI / float(nPoints-1.0)) * findgen(nPoints)
x = 0.5 + 0.2 * cos(points)
y = 0.5 + 0.2 * sin(points)
pl = polygon(x*500.0,y*500.0, '-b2', fill_background=0, /data, target=io)
pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)
end
```

Now the problems:
1) If I select the circle (blue data) and I try to move it, I can only do that for specific mouse positions. These positions are not consistent with the line. Is this a feature or a bug?
2) I cannot move the line at all except if I try to rescale it. Is this a feature or a bug?
3) With the above code I get inconsistent math error messages that I cannot pin down to a specific operation. The errors look like:

```
% Program caused arithmetic error: Floating overflow
% Program caused arithmetic error: Floating illegal operand
% Detected at IDLITVISDATASPACE::DRAW 2385 C:\Program
Files\Exelis\IDL84\IDL84\lib\itools\framework\idlitvisdatasp ace__define.pro
% Program caused arithmetic error: Floating overflow
% Detected at IDLITVISDATASPACE::DRAW 2385 C:\Program
Files\Exelis\IDL84\IDL84\lib\itools\framework\idlitvisdatasp ace__define.pro
% Program caused arithmetic error: Floating illegal operand
% Detected at IDLITVISDATASPACE::DRAW 2385 C:\Program
Files\Exelis\IDL84\IDL84\lib\itools\framework\idlitvisdatasp ace__define.pro
```

And it happens when I move the mouse to about the center of the image and click. *If* it happens once, then it keeps on throwing errors when moving the mouse on the image.

Notice that I set:
!DEBUG_PROCESS_EVENTS = 0
!EXCEPT = 2

before testing this code.
I reset IDL before I test.
The error happens with a frequency of ~1/10 to 1/20.

Last thing:
IDL> !version

{
   "ARCH": "x86_64",
   "OS": "Win32",
   "OS_FAMILY": "Windows",
   "OS_NAME": "MicrosoftWindows",
   "RELEASE": "8.4",
   "BUILD_DATE": "Sep272014",
   "MEMORY_BITS": 64,
   "FILE_OFFSET_BITS": 64
}

Cheers,
Helder

PS1: the references to the code I got by using !EXCEPT = 2
PS2: !EXCEPT = 2 makes IDL ~5% slower! For more details see:
 http://www.exelisvis.com/docs/error_handling_system_va.html# sysvars_272074529_1002623

Subject: Re: FG-arithmetic error and moving objects
Posted by Matthew Argall on Tue, 03 Mar 2015 13:10:33 GMT
View Forum Message <> Reply to Message

After a .reset, I copy-pasted your code and got similar errors, but from the IDLitSymbol object. The error occurs always when I select the image (mouse motion events cause the error), but do not occur for the circle unless I drag it over to the edge of the window (which was hard to do).

% Program caused arithmetic error: Floating divide by 0
% Program caused arithmetic error: Floating illegal operand
% Detected at IDLITSYMBOL::DRAW   1
/Applications/exelis/idl82/lib/itools/components/idlitsymbol __define.pro

The only thing I can image that causes this are the symbols that indicate where stretches and rotates can occur when a graphic is selected. Perhaps the smallness of these symbols within the data space causes the errors?

As for the other items,

1) I cannot move the circle unless the mouse is near the outlining selection border

2) I cannot translate the line at all using the mouse (although I can with the Translate method).

```
IDL> help, !version
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'darwin'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'Mac OS X'
   RELEASE         STRING    '8.2'
   BUILD_DATE      STRING    'Apr 10 2012'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

Subject: Re: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Tue, 03 Mar 2015 13:39:54 GMT
View Forum Message <> Reply to Message

On Tuesday, March 3, 2015 at 2:10:36 PM UTC+1, Matthew Argall wrote:
> After a .reset, I copy-pasted your code and got similar errors, but from the IDLitSymbol object. The error occurs always when I select the image (mouse motion events cause the error), but do not occur for the circle unless I drag it over to the edge of the window (which was hard to do).
>
>
> % Program caused arithmetic error: Floating divide by 0
> % Program caused arithmetic error: Floating illegal operand
> % Detected at IDLITSYMBOL::DRAW 1 /Applications/exelis/idl82/lib/itools/components/idlitsymbol __define.pro
>
>
> The only thing I can image that causes this are the symbols that indicate where stretches and rotates can occur when a graphic is selected. Perhaps the smallness of these symbols within the data space causes the errors?
>
>
> As for the other items,
>
> 1) I cannot move the circle unless the mouse is near the outlining selection border
> 2) I cannot translate the line at all using the mouse (although I can with the Translate method).
>
>
> IDL> help, !version
> ** Structure !VERSION, 8 tags, length=104, data length=100:
> ARCH STRING 'x86_64'
> OS STRING 'darwin'
> OS_FAMILY STRING 'unix'

>   OS_NAME   STRING   'Mac OS X'
>   RELEASE   STRING   '8.2'
>   BUILD_DATE   STRING   'Apr 10 2012'
>   MEMORY_BITS   INT   64
>   FILE_OFFSET_BITS
>   INT   64

Hi,
maybe at the end of the init method of idlitsymbol__define.pro one should change the line:

    ; Mac sometimes throws floating underflows...

to:

    ; Mac and windows sometimes throw floating underflows, or floating overflow...

Cheers, Helder

Subject: Re: FG-arithmetic error and moving objects
Posted by David Fanning on Tue, 03 Mar 2015 14:00:11 GMT
View Forum Message <> Reply to Message

Helder writes:

> Now the problems:
> 1) If I select the circle (blue data) and I try to move it, I can only do that for specific mouse positions. These positions are not consistent with the line. Is this a feature or a bug?
> 2) I cannot move the line at all except if I try to rescale it. Is this a feature or a bug?
> 3) With the above code I get inconsistent math error messages that I cannot pin down to a specific operation.

As I pretty much expected, your example doesn't work at all in the version of IDL (8.2.3) I have. On the other hand, it doesn't throw errors when I move the mouse. ;-)

What it does do, and this made me think I was either going crazy or the coffee was especially strong this morning, is cause faint (you might say subliminal) yellow copies of the circle to appear at random locations in the window, even when I stop moving the mouse! It is one of the damnedest things I've ever seen in IDL. :-)

Oh, now as I stare at the window, I occasionally see a cyan version of the line popping into momentary existence, too. This must be what it is like to chase subatomic particles in a bubble chamber. I thought at first this must have something to do with my glasses, but it happens when I hold my head completely still. It is like a memory of the two objects flickering into and out of existence. Fascinating!

Cheers,

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

Subject: Re: FG-arithmetic error and moving objects
Posted by wlandsman on Tue, 03 Mar 2015 14:07:17 GMT
View Forum Message <> Reply to Message

On Tuesday, March 3, 2015 at 9:00:15 AM UTC-5, David Fanning wrote:
>
> Oh, now as I stare at the window, I occasionally see a cyan version of
> the line popping into momentary existence, too.

Is the line gold and white or blue and black?

Subject: Re: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Tue, 03 Mar 2015 14:15:57 GMT
View Forum Message <> Reply to Message

On Tuesday, March 3, 2015 at 3:07:19 PM UTC+1, wlandsman wrote:
> On Tuesday, March 3, 2015 at 9:00:15 AM UTC-5, David Fanning wrote:
>>
>> Oh, now as I stare at the window, I occasionally see a cyan version of
>> the line popping into momentary existence, too.
>
> Is the line gold and white or blue and black?

I +1 this!

h

Subject: Re: FG-arithmetic error and moving objects
Posted by Yngvar Larsen on Tue, 03 Mar 2015 14:28:26 GMT
View Forum Message <> Reply to Message

Sounds like some kind of event loop bug in combination with the FG equivalent of this:

http://www.exelisvis.com/docs/DEVICE_Procedure.html#devices_ 517620971_499029

On Tuesday, 3 March 2015 15:00:15 UTC+1, David Fanning wrote:

> What it does do, and this made me think I was either going crazy or the
> coffee was especially strong this morning, is cause faint (you might say
> subliminal) yellow copies of the circle to appear at random locations in
> the window, even when I stop moving the mouse! It is one of the
> damnedest things I've ever seen in IDL. :-)
>
> Oh, now as I stare at the window, I occasionally see a cyan version of
> the line popping into momentary existence, too. This must be what it is
> like to chase subatomic particles in a bubble chamber. I thought at
> first this must have something to do with my glasses, but it happens
> when I hold my head completely still. It is like a memory of the two
> objects flickering into and out of existence. Fascinating!

--
Yngvar

Subject: Re: FG-arithmetic error and moving objects
Posted by Matthew Argall on Tue, 03 Mar 2015 14:37:52 GMT
View Forum Message <> Reply to Message

> yellow copies of the circle to appear at random locations in
> the window, even when I stop moving the mouse!
>
> Oh, now as I stare at the window, I occasionally see a cyan version of
> the line popping into momentary existence, too.

Recently, I have noticed that the Symbol and Polyline objects in 8.2 do not obey the refresh state of the window. I can say

win -> Refresh, /DISABLE

then call the SetData method of a Symbol or a Polyline and the window will update automatically, first at the wrong scaling, then at the proper scaling.

Since a version changes of 0.0.1 any time after 8.0 make previous versions of IDL obsolete, I have not brought it up. I have also not brought up the fact that the IDL help pages are not backwards compatible, which is extremely frustrating -- it is impossible to make an Axis() in versions prior to 8.2.2 without scrolling all the way to the bottom of the page and reading "The old LOCATION behavior (with a 3-element vector) is no longer documented, but will continue to work" in the version history.

Subject: Re: FG-arithmetic error and moving objects

Posted by Dick Jackson on Tue, 03 Mar 2015 16:52:49 GMT
View Forum Message <> Reply to Message

Hi Helder,

I will bet that Exelis folks are taking notes from this (Hi, Chris!), so I'll add some more to your report.

On Tuesday, 3 March 2015 04:23:57 UTC-8, Helder wrote:
> Hi,
> as already discussed in previous posts, I'm having trouble with annotations on images. I have a minimal example to show three problems.

> Now the problems:
> 1) If I select the circle (blue data) and I try to move it, I can only do that for specific mouse positions. These positions are not consistent with the line. Is this a feature or a bug?
> 2) I cannot move the line at all except if I try to rescale it. Is this a feature or a bug?
> 3) With the above code I get inconsistent math error messages that I cannot pin down to a specific operation. The errors look like:
>
> % Program caused arithmetic error: Floating overflow
> % Program caused arithmetic error: Floating illegal operand
> % Detected at IDLITVISDATASPACE::DRAW 2385 C:\Program Files\Exelis\IDL84\IDL84\lib\itools\framework\idlitvisdatasp ace__define.pro
> [etc.]

Here's my setup (Windows 7, same IDL as you):
```
IDL> help,!version
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'Win32'
   OS_FAMILY       STRING    'Windows'
   OS_NAME         STRING    'Microsoft Windows'
   RELEASE         STRING    '8.4'
   BUILD_DATE      STRING    'Sep 27 2014'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

I do see the problems that you describe (but, as Matthew, from the IDLitSymbol object), and further: if I resize the red line, then things change, and we're now in a 3-D world. I can grab the image and end up rotating the whole space around. What is also revealed then is that the red line is no longer on the Z=0 plane. At the command line (I just pasted the lines from your procedure in there), I can do this:

(before resizing red line)
```
IDL> pl.getdata,xx,yy,zz
IDL> print,xx,yy,zz
      125.00000       375.00000
```

```
      125.00000       375.00000
     0.00000000      0.00000000
```

(after resizing red line)
IDL> pl.getdata,xx,yy,zz
IDL> print,xx,yy,zz
```
      125.00000       414.00000
      125.00000       180.00000
     0.00000000       1.7147294
```

That's odd, but it explains why we can suddenly rotate the space. And it's then hard to do any of the usual things with the annotations, and that's when I start seeing the arithmetic errors. (It's a good thing Ctrl-Z works at this point to Undo changes... quite dependably, but not perfectly, I find)

And to David: I'm going to guess that what you're seeing is yellow complementary afterimages. (it's all in your head! :-) Since the world is suddenly interested in our visual system this past week, I'll toss out this barely-relevant (but fascinating) link:
https://en.wikipedia.org/wiki/Lilac_chaser

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada --- http://www.d-jackson.com

Subject: Re: FG-arithmetic error and moving objects
Posted by David Fanning on Tue, 03 Mar 2015 17:29:59 GMT
View Forum Message <> Reply to Message

Dick Jackson writes:

> And to David: I'm going to guess that what you're seeing is yellow complementary afterimages. (it's all in your head! :-) Since the world is suddenly interested in our visual system this past week, I'll toss out this barely-relevant (but fascinating) link:
> https://en.wikipedia.org/wiki/Lilac_chaser

Totally weird. Is it coincidence that the sun is shining and it is
snowing simultaneously here? :-)

Cheers,

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

Subject: Re: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Tue, 03 Mar 2015 20:06:21 GMT
View Forum Message <> Reply to Message

> I do see the problems that you describe (but, as Matthew, from the IDLitSymbol object), and further: if I resize the red line, then things change, and we're now in a 3-D world. I can grab the image and end up rotating the whole space around. What is also revealed then is that the red line is no longer on the Z=0 plane. At the command line (I just pasted the lines from your procedure in there), I can do this:
>
> (before resizing red line)
> IDL> pl.getdata,xx,yy,zz
> IDL> print,xx,yy,zz
> 125.00000 375.00000
> 125.00000 375.00000
> 0.00000000 0.00000000
>
> (after resizing red line)
> IDL> pl.getdata,xx,yy,zz
> IDL> print,xx,yy,zz
> 125.00000 414.00000
> 125.00000 180.00000
> 0.00000000 1.7147294
>
> That's odd, but it explains why we can suddenly rotate the space. And it's then hard to do any of the usual things with the annotations, and that's when I start seeing the arithmetic errors. (It's a good thing Ctrl-Z works at this point to Undo changes... quite dependably, but not perfectly, I find)
>
> And to David: I'm going to guess that what you're seeing is yellow complementary afterimages. (it's all in your head! :-) Since the world is suddenly interested in our visual system this past week, I'll toss out this barely-relevant (but fascinating) link:
> https://en.wikipedia.org/wiki/Lilac_chaser
>
> Cheers,
> -Dick
>
> Dick Jackson Software Consulting Inc.
> Victoria, BC, Canada --- http://www.d-jackson.com

Hi Dick,
we've got that bug of the 2d->3d when resizing a line a few days ago:
https://groups.google.com/d/msg/comp.lang.idl-pvwave/fmtuiHN 031M/m1tNLK1YJQUJ

or if you prefer:

http://idl.marchetto.de/bug-and-fix-for-a-2d-image-with-a-li ne-rotates-in-3d/

Thanks to Chris @ Exelis the fix was pretty easy and came quickly!

Cheers,
Helder

PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.

Subject: Re: FG-arithmetic error and moving objects
Posted by David Fanning on Tue, 03 Mar 2015 20:20:16 GMT
View Forum Message <> Reply to Message

Helder writes:

> PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>

A +1 for this.

Cheers,

David

P.S. Let's just say this is not the most surprising thing I've heard
today. :-)

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

Subject: Re: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Wed, 04 Mar 2015 08:44:18 GMT
View Forum Message <> Reply to Message

On Tuesday, March 3, 2015 at 9:20:20 PM UTC+1, David Fanning wrote:
> Helder writes:
>
>> PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>>
>
> A +1 for this.
>
> Cheers,

>
> David
>
> P.S. Let's just say this is not the most surprising thing I've heard
> today. :-)
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Hi,
frustrated by these errors I just found one workaround and I understood where 4-arrows mouse cursor used to move objects appears.

First the (partial) workaround for not being able to move the line. Easy. Just use
pg2 = polygon([0.25,0.75,0.25]*500.0,[0.25,0.75,0.25]*500.0, '-r2', /data, target=io)
instead of
pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)

This only works only partially because now you cannot make the line horizontal or vertical...

As for the moving of objects, I realized (and this is obvious with the polygon line) that polygons can be moved when the mouse is on the edges of the minimum bounding box.

As far as I'm concerned, point 1) in the original post is clear: it's not a bug, it's a feature. Objects movement respond to the minimum bounding box, not the edges of the polygon.

[Note: I switched some months ago to FG graphics. It's easier to make nice plots/images, to save them and I like the idea of not having to handle the movements of complex objects myself. I'm slowly realizing, at my own loss of time, that appearances have tricked me. Direct graphics has been tested for over ... well before I even knew IDL existed. And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].

As far as the polyline to polygon fix, I'm not happy with this. I NEED lines to take any direction, including horizontal and vertical.

And the arithmetic error can be hidden to the user, but it is still suspicious of something going badly wrong.

Cheers,
Helder

Subject: Re: FG-arithmetic error and moving objects
Posted by David Fanning on Wed, 04 Mar 2015 14:40:13 GMT

Helder writes:

> And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].

The good news is, things have greatly improved. When I had this job, all
I could get to work correctly was a simple line plot:

http://www.idlcoyote.com/ng_tips/onion.php

It's a good job. It will keep you employed for years. ;-)

Cheers,

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

Subject: Re: FG-arithmetic error and moving objects
Posted by chris_torrence@NOSPAM on Thu, 05 Mar 2015 16:42:26 GMT

On Wednesday, March 4, 2015 at 1:44:23 AM UTC-7, Helder wrote:
> On Tuesday, March 3, 2015 at 9:20:20 PM UTC+1, David Fanning wrote:
>> Helder writes:
>>
>>> PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>>>
>>
>> A +1 for this.
>>
>> Cheers,
>>
>> David
>>
>> P.S. Let's just say this is not the most surprising thing I've heard
>> today. :-)
>>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.

>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>
> Hi,
> frustrated by these errors I just found one workaround and I understood where 4-arrows mouse cursor used to move objects appears.
>
> First the (partial) workaround for not being able to move the line. Easy. Just use
> pg2 = polygon([0.25,0.75,0.25]*500.0,[0.25,0.75,0.25]*500.0, '-r2', /data, target=io)
> instead of
> pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)
>
> This only works only partially because now you cannot make the line horizontal or vertical...
>
> As for the moving of objects, I realized (and this is obvious with the polygon line) that polygons can be moved when the mouse is on the edges of the minimum bounding box.
>
> As far as I'm concerned, point 1) in the original post is clear: it's not a bug, it's a feature. Objects movement respond to the minimum bounding box, not the edges of the polygon.
>
> [Note: I switched some months ago to FG graphics. It's easier to make nice plots/images, to save them and I like the idea of not having to handle the movements of complex objects myself. I'm slowly realizing, at my own loss of time, that appearances have tricked me. Direct graphics has been tested for over ... well before I even knew IDL existed. And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].
>
> As far as the polyline to polygon fix, I'm not happy with this. I NEED lines to take any direction, including horizontal and vertical.
>
> And the arithmetic error can be hidden to the user, but it is still suspicious of something going badly wrong.
>
> Cheers,
> Helder

Hi Helder,

Don't give up!

Regarding your issues, the main problem is that we purposely limited the translation, scaling, and rotation when annotations are in the dataspace. The assumption is that if your annotation is in "data" coordinates then you don't want it to move. Perhaps we need to re-think this limitation...

Regarding the floating-point arithmetic errors, *sometimes* these are legitimate errors but most of the time they are just harmless noise that can be safely ignored. There are lots of places in the graphics code where a number will get rounded off to zero, or some calculation will make another number go to Infinity or NaN, and then IDL will dutifully report the exception. If you look through the graphics pro code, you will find many instances where we just quietly swallow those. So

unless you actually see some visual "badness", I would just ignore them.

Cheers,

Chris
Exelis
p.s. keep posting your questions and feedback. We really do listen!

Subject: Re: FG-arithmetic error and moving objects
Posted by Helder Marchetto on Thu, 05 Mar 2015 20:50:28 GMT
View Forum Message <> Reply to Message

On Thursday, March 5, 2015 at 5:42:29 PM UTC+1, Chris Torrence wrote:
> On Wednesday, March 4, 2015 at 1:44:23 AM UTC-7, Helder wrote:
>> On Tuesday, March 3, 2015 at 9:20:20 PM UTC+1, David Fanning wrote:
>>> Helder writes:
>>>
>>>> PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>>>>
>>>
>>> A +1 for this.
>>>
>>> Cheers,
>>>
>>> David
>>>
>>> P.S. Let's just say this is not the most surprising thing I've heard
>>> today. :-)
>>>
>>> --
>>> David Fanning, Ph.D.
>>> Fanning Software Consulting, Inc.
>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>>
>> Hi,
>> frustrated by these errors I just found one workaround and I understood where 4-arrows mouse cursor used to move objects appears.
>>
>> First the (partial) workaround for not being able to move the line. Easy. Just use
>> pg2 = polygon([0.25,0.75,0.25]*500.0,[0.25,0.75,0.25]*500.0, '-r2', /data, target=io)
>> instead of
>> pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)
>>
>> This only works only partially because now you cannot make the line horizontal or vertical...
>>

>> As for the moving of objects, I realized (and this is obvious with the polygon line) that polygons can be moved when the mouse is on the edges of the minimum bounding box.
>>
>> As far as I'm concerned, point 1) in the original post is clear: it's not a bug, it's a feature. Objects movement respond to the minimum bounding box, not the edges of the polygon.
>>
>> [Note: I switched some months ago to FG graphics. It's easier to make nice plots/images, to save them and I like the idea of not having to handle the movements of complex objects myself. I'm slowly realizing, at my own loss of time, that appearances have tricked me. Direct graphics has been tested for over ... well before I even knew IDL existed. And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].
>>
>> As far as the polyline to polygon fix, I'm not happy with this. I NEED lines to take any direction, including horizontal and vertical.
>>
>> And the arithmetic error can be hidden to the user, but it is still suspicious of something going badly wrong.
>>
>> Cheers,
>> Helder
>
> Hi Helder,
>
> Don't give up!
>
> Regarding your issues, the main problem is that we purposely limited the translation, scaling, and rotation when annotations are in the dataspace. The assumption is that if your annotation is in "data" coordinates then you don't want it to move. Perhaps we need to re-think this limitation...
>
> Regarding the floating-point arithmetic errors, *sometimes* these are legitimate errors but most of the time they are just harmless noise that can be safely ignored. There are lots of places in the graphics code where a number will get rounded off to zero, or some calculation will make another number go to Infinity or NaN, and then IDL will dutifully report the exception. If you look through the graphics pro code, you will find many instances where we just quietly swallow those. So unless you actually see some visual "badness", I would just ignore them.
>
> Cheers,
>
> Chris
> Exelis
> p.s. keep posting your questions and feedback. We really do listen!

Hi Chris,
thanks for your answer.
It's just half time and my favorite team is neither losing nor winning, so I take a few minutes to answer.
I'll disregard the floating-point arithmetic errors. It's just something that does not go unnoticed. You might want to fit that in the documentation (if it's already inside... sorry, I'm a distracted

reader).

As far as the data space and the translations, here are my 2 cents: I've already looked quite some time ago into how to make annotations insensitive to mouse (not movable) and you've just posted the answer also elsewhere (return, 0 to handling functions).
My suggestion: why not use keywords to lock annotations, maybe with the /lock keyword? Then one can have a consistent (identical) behavior for normal, relative, device and data coordinates.

Maybe I should also say "why" I find the freedom of moving annotations in data space important. I want to place annotations (areas or lines to use to calculate intensity profiles) on images. The nice thing about FG, is that one can zoom easily with the mouse wheel and position the annotation accurately. If the annotation was defined in data space, it will "stay" where it was when zooming. This is good, but once I zoomed in, I would like to adjust the position.

I hope you get the point. If not I can build some example. Of course one can find workarounds and I've been doing this since years with direct graphics, but I find that the added value of FG should just be to handle processes as describe above in a clean and direct way without having to deal with the dirty stuff...

Cheers,
Helder

PS: Football is back in a few minutes. I'll IDL tomorrow again.

Subject: Re: FG-arithmetic error and moving objects
Posted by jimuba on Thu, 16 Apr 2015 00:38:00 GMT
View Forum Message <> Reply to Message

On Thursday, March 5, 2015 at 1:50:33 PM UTC-7, Helder wrote:
> On Thursday, March 5, 2015 at 5:42:29 PM UTC+1, Chris Torrence wrote:
>> On Wednesday, March 4, 2015 at 1:44:23 AM UTC-7, Helder wrote:
>>> On Tuesday, March 3, 2015 at 9:20:20 PM UTC+1, David Fanning wrote:
>>>> Helder writes:
>>>>
>>>> > PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>>>> >
>>>>
>>>> A +1 for this.
>>>>
>>>> Cheers,
>>>>
>>>> David
>>>>
>>>> P.S. Let's just say this is not the most surprising thing I've heard
>>>> today. :-)
>>>>

>>>> --
>>>> David Fanning, Ph.D.
>>>> Fanning Software Consulting, Inc.
>>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>>>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>>>
>>> Hi,
>>> frustrated by these errors I just found one workaround and I understood where 4-arrows mouse cursor used to move objects appears.
>>>
>>> First the (partial) workaround for not being able to move the line. Easy. Just use
>>> pg2 = polygon([0.25,0.75,0.25]*500.0,[0.25,0.75,0.25]*500.0, '-r2', /data, target=io)
>>> instead of
>>> pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)
>>>
>>> This only works only partially because now you cannot make the line horizontal or vertical...
>>>
>>> As for the moving of objects, I realized (and this is obvious with the polygon line) that polygons can be moved when the mouse is on the edges of the minimum bounding box.
>>>
>>> As far as I'm concerned, point 1) in the original post is clear: it's not a bug, it's a feature. Objects movement respond to the minimum bounding box, not the edges of the polygon.
>>>
>>> [Note: I switched some months ago to FG graphics. It's easier to make nice plots/images, to save them and I like the idea of not having to handle the movements of complex objects myself. I'm slowly realizing, at my own loss of time, that appearances have tricked me. Direct graphics has been tested for over ... well before I even knew IDL existed. And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].
>>>
>>> As far as the polyline to polygon fix, I'm not happy with this. I NEED lines to take any direction, including horizontal and vertical.
>>>
>>> And the arithmetic error can be hidden to the user, but it is still suspicious of something going badly wrong.
>>>
>>> Cheers,
>>> Helder
>>
>> Hi Helder,
>>
>> Don't give up!
>>
>> Regarding your issues, the main problem is that we purposely limited the translation, scaling, and rotation when annotations are in the dataspace. The assumption is that if your annotation is in "data" coordinates then you don't want it to move. Perhaps we need to re-think this limitation...
>>
>> Regarding the floating-point arithmetic errors, *sometimes* these are legitimate errors but most of the time they are just harmless noise that can be safely ignored. There are lots of places

in the graphics code where a number will get rounded off to zero, or some calculation will make another number go to Infinity or NaN, and then IDL will dutifully report the exception. If you look through the graphics pro code, you will find many instances where we just quietly swallow those. So unless you actually see some visual "badness", I would just ignore them.
>>
>> Cheers,
>>
>> Chris
>> Exelis
>> p.s. keep posting your questions and feedback. We really do listen!
>
> Hi Chris,
> thanks for your answer.
> It's just half time and my favorite team is neither losing nor winning, so I take a few minutes to answer.
> I'll disregard the floating-point arithmetic errors. It's just something that does not go unnoticed. You might want to fit that in the documentation (if it's already inside... sorry, I'm a distracted reader).
>
> As far as the data space and the translations, here are my 2 cents: I've already looked quite some time ago into how to make annotations insensitive to mouse (not movable) and you've just posted the answer also elsewhere (return, 0 to handling functions).
> My suggestion: why not use keywords to lock annotations, maybe with the /lock keyword? Then one can have a consistent (identical) behavior for normal, relative, device and data coordinates.
>
> Maybe I should also say "why" I find the freedom of moving annotations in data space important. I want to place annotations (areas or lines to use to calculate intensity profiles) on images. The nice thing about FG, is that one can zoom easily with the mouse wheel and position the annotation accurately. If the annotation was defined in data space, it will "stay" where it was when zooming. This is good, but once I zoomed in, I would like to adjust the position.
>
> I hope you get the point. If not I can build some example. Of course one can find workarounds and I've been doing this since years with direct graphics, but I find that the added value of FG should just be to handle processes as describe above in a clean and direct way without having to deal with the dirty stuff...
>
> Cheers,
> Helder
>
> PS: Football is back in a few minutes. I'll IDL tomorrow again.

Hi,

It seems that the problem is that a 2-vertex polyline graphic is missing the translate and rotate manipulator handles that you normally get with a polyline that has 3 or more vertices.

With this in mind, a possible workaround to the problem is to add a 3rd vertex that duplicates the 2nd vertex to your polyline. For example:

```
img = image(dist(400,400))
p2 = polyline([100,300],[200,300],'-r2',/data,target=img)
p3 = polyline([100,300,300],[100,200,200],'-b2',/data, target=img)
```

Then when you select the blue line, the rectangular manipulator frame will appear, giving you the ability to move the line without altering the shape or slope. You can also rotate the line without changing it's shape.

Regards,
Jim
Exelis

Subject: Re: FG-arithmetic error and moving objects
Posted by jimuba on Thu, 16 Apr 2015 00:43:36 GMT
View Forum Message <> Reply to Message

On Thursday, March 5, 2015 at 1:50:33 PM UTC-7, Helder wrote:
> On Thursday, March 5, 2015 at 5:42:29 PM UTC+1, Chris Torrence wrote:
>> On Wednesday, March 4, 2015 at 1:44:23 AM UTC-7, Helder wrote:
>>> On Tuesday, March 3, 2015 at 9:20:20 PM UTC+1, David Fanning wrote:
>>>> Helder writes:
>>>>
>>>> > PS: looks like I got a FG-bug-magnet under my keyboard... I just keep on bumping onto these sort of stuff.
>>>> >
>>>>
>>>> A +1 for this.
>>>>
>>>> Cheers,
>>>>
>>>> David
>>>>
>>>> P.S. Let's just say this is not the most surprising thing I've heard
>>>> today. :-)
>>>>
>>>> --
>>>> David Fanning, Ph.D.
>>>> Fanning Software Consulting, Inc.
>>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>>>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>>>
>>> Hi,
>>> frustrated by these errors I just found one workaround and I understood where 4-arrows mouse cursor used to move objects appears.
>>>
>>> First the (partial) workaround for not being able to move the line. Easy. Just use

>>> pg2 = polygon([0.25,0.75,0.25]*500.0,[0.25,0.75,0.25]*500.0, '-r2', /data, target=io)
>>> instead of
>>> pl = polyline([0.25,0.75]*500.0,[0.25,0.75]*500.0, '-r2', /data, target=io)
>>>
>>> This only works only partially because now you cannot make the line horizontal or vertical...
>>>
>>> As for the moving of objects, I realized (and this is obvious with the polygon line) that polygons can be moved when the mouse is on the edges of the minimum bounding box.
>>>
>>> As far as I'm concerned, point 1) in the original post is clear: it's not a bug, it's a feature. Objects movement respond to the minimum bounding box, not the edges of the polygon.
>>>
>>> [Note: I switched some months ago to FG graphics. It's easier to make nice plots/images, to save them and I like the idea of not having to handle the movements of complex objects myself. I'm slowly realizing, at my own loss of time, that appearances have tricked me. Direct graphics has been tested for over ... well before I even knew IDL existed. And I have the feeling I'm getting to be the official bug-finder* of FG (I say this without pretenses, only frustration)].
>>>
>>> As far as the polyline to polygon fix, I'm not happy with this. I NEED lines to take any direction, including horizontal and vertical.
>>>
>>> And the arithmetic error can be hidden to the user, but it is still suspicious of something going badly wrong.
>>>
>>> Cheers,
>>> Helder
>>
>> Hi Helder,
>>
>> Don't give up!
>>
>> Regarding your issues, the main problem is that we purposely limited the translation, scaling, and rotation when annotations are in the dataspace. The assumption is that if your annotation is in "data" coordinates then you don't want it to move. Perhaps we need to re-think this limitation...
>>
>> Regarding the floating-point arithmetic errors, *sometimes* these are legitimate errors but most of the time they are just harmless noise that can be safely ignored. There are lots of places in the graphics code where a number will get rounded off to zero, or some calculation will make another number go to Infinity or NaN, and then IDL will dutifully report the exception. If you look through the graphics pro code, you will find many instances where we just quietly swallow those. So unless you actually see some visual "badness", I would just ignore them.
>>
>> Cheers,
>>
>> Chris
>> Exelis
>> p.s. keep posting your questions and feedback. We really do listen!
>

> Hi Chris,
> thanks for your answer.
> It's just half time and my favorite team is neither losing nor winning, so I take a few minutes to answer.
> I'll disregard the floating-point arithmetic errors. It's just something that does not go unnoticed. You might want to fit that in the documentation (if it's already inside... sorry, I'm a distracted reader).
>
> As far as the data space and the translations, here are my 2 cents: I've already looked quite some time ago into how to make annotations insensitive to mouse (not movable) and you've just posted the answer also elsewhere (return, 0 to handling functions).
> My suggestion: why not use keywords to lock annotations, maybe with the /lock keyword? Then one can have a consistent (identical) behavior for normal, relative, device and data coordinates.
>
> Maybe I should also say "why" I find the freedom of moving annotations in data space important. I want to place annotations (areas or lines to use to calculate intensity profiles) on images. The nice thing about FG, is that one can zoom easily with the mouse wheel and position the annotation accurately. If the annotation was defined in data space, it will "stay" where it was when zooming. This is good, but once I zoomed in, I would like to adjust the position.
>
> I hope you get the point. If not I can build some example. Of course one can find workarounds and I've been doing this since years with direct graphics, but I find that the added value of FG should just be to handle processes as describe above in a clean and direct way without having to deal with the dirty stuff...
>
> Cheers,
> Helder
>
> PS: Football is back in a few minutes. I'll IDL tomorrow again.

Hi,

It seems that the problem is that a 2-vertex polyline graphic is missing the translate and rotate manipulator handles that you normally get with a polyline that has 3 or more vertices.

With this in mind, a possible workaround to the problem is to add a 3rd vertex that duplicates the 2nd vertex to your polyline. For example:

```
img = image(dist(400,400))
p2 = polyline([100,300],[200,300],'-r2',/data,target=img)
p3 = polyline([100,300,300],[100,200,200],'-b2',/data, target=img)
```

Then when you select the blue line, the rectangular manipulator frame will appear, giving you the ability to move the line without altering the shape or slope. You can also rotate the line without changing its shape.

(FYI, a bug report (IDL-69358) has been filed for this issue.)

Regards,
Jim
Exelis