
Subject: IDL equivalent to MATLAB's linkaxes

Posted by [Gordon Farquharson](#) on Fri, 02 Oct 2015 19:48:54 GMT

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

Hi All

Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something one would have to implement by updating each figure's xrange and yrange properties when some zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image function [2] be the best way to go?

Gordon

[1] <http://www.mathworks.com/help/matlab/ref/linkaxes.html?refres=1>

[2] <http://www.exelisvis.com/docs/graphicseventhandler.html>

Subject: Re: IDL equivalent to MATLAB's linkaxes

Posted by [Helder Marchetto](#) on Fri, 02 Oct 2015 20:31:58 GMT

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

On Friday, October 2, 2015 at 9:48:57 PM UTC+2, Gordon Farquharson wrote:

> Hi All

>

> Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something one would have to implement by updating each figure's xrange and yrange properties when some zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image function [2] be the best way to go?

>

> Gordon

>

> [1] <http://www.mathworks.com/help/matlab/ref/linkaxes.html?refres=1>

> [2] <http://www.exelisvis.com/docs/graphicseventhandler.html>

Hi,

I've always wanted this and could not realize it 100%. The reason is the following: when you use the event handler of FG, you have to write functions that catch the event (for instance "mouse wheel" for zoom), do whatever processing and then if you end the function with:

1) return, 0: then IDL won't process the specific event at all (no zooming will be performed)

2) return, 1: then IDL will process the event normally (zooming will be performed)

So now you have two ways out of this:

a) make use of option 1) from above and process the event yourself. This means zoom the image yourself.

b) make use of option 2) and with a delay (timer) do some post-event processing.

I tried to do this for the zoom for a completely different problem (moving annotation on the image with the image when zooming and panning) and I found in the course of this another problem that I reported here: <http://idl.marchetto.de/annotations-on-images/>

Nevertheless you might still find it useful. For example, one of the most important things you need to know to handle the zoom is the zooming factor. In FG this is given by:

$\text{zoomFactor} = (\text{Delta} > 0) ? 1.25^d : 1/1.25^d$

where Delta is given in the call to the function:

function WheelEvent, oWin, xPos, yPos, Delta, KeyMods

In case you're going for the pan option too, what you could do, is process the mouse up option. First make notice if you're dealing with pan (mouse pressed + move), then in the mouse up function, get the new axes values and send them over to the other plot.

Hope it helps and if you find a better solution, plz let me know.

Regards,
Helder

Subject: Re: IDL equivalent to MATLAB's linkaxes

Posted by [Gordon Farquharson](#) on Mon, 05 Oct 2015 21:28:22 GMT

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

On Friday, October 2, 2015 at 1:32:08 PM UTC-7, Helder wrote:

> On Friday, October 2, 2015 at 9:48:57 PM UTC+2, Gordon Farquharson wrote:

>> Hi All

>>

>> Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something one would have to implement by updating each figure's xrange and yrange properties when some zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image function [2] be the best way to go?

>>

>> Gordon

>>

>> [1] <http://www.mathworks.com/help/matlab/ref/linkaxes.html?refre sh=true>

>> [2] <http://www.exelisvis.com/docs/graphicseventhandler.html>

>

> Hi,

> I've always wanted this and could not realize it 100%. The reason is the following: when you use the event handler of FG, you have to write functions that catch the event (for instance "mouse wheel" for zoom), do whatever processing and then if you end the function with:

>

> 1) return, 0: then IDL won't process the specific event at all (no zooming will be performed)

>

> 2) return, 1: then IDL will process the event normally (zooming will be performed)

>

> So now you have two ways out of this:
>
> a) make use of option 1) from above and process the event yourself. This means zoom the image yourself.
>
> b) make use of option 2) and with a delay (timer) do some post-event processing.
>
> I tried to do this for the zoom for a completely different problem (moving annotation on the image with the image when zooming and panning) and I found in the course of this another problem that I reported here: <http://idl.marchetto.de/annotations-on-images/>
> Nevertheless you might still find it useful. For example, one of the most important things you need to know to handle the zoom is the zooming factor. In FG this is given by:
> zoomFactor = (Delta > 0) ? 1.25d : 1/1.25d
> where Delta is given in the call to the function:
> function WheelEvent, oWin, xPos, yPos, Delta, KeyMods
>
> In case you're going for the pan option too, what you could do, is process the mouse up option. First make notice if you're dealing with pan (mouse pressed + move), then in the mouse up function, get the new axes values and send them over to the other plot.
>
> Hope it helps and if you find a better solution, plz let me know.
>
> Regards,
> Helder

Hi Helder

Thanks very much. This information does help. It turns out that I won't be able to invest the time in implementing something now, but I'll let you if I do ever come up with a solution.

Gordon

Subject: Re: IDL equivalent to MATLAB's linkaxes
Posted by [braveheart197315](#) on Wed, 14 Oct 2015 09:12:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, 2 October 2015 21:32:08 UTC+1, Helder wrote:

> On Friday, October 2, 2015 at 9:48:57 PM UTC+2, Gordon Farquharson wrote:
>> Hi All
>>
>> Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something one would have to implement by updating each figure's xrange and yrange properties when some zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image function [2] be the best way to go?
>>
>> Gordon
>>

```
>> [1] http://www.mathworks.com/help/matlab/ref/linkaxes.html?refre sh=true
>> [2] http://www.exelisvis.com/docs/graphicseventhandler.html
>
> Hi,
> I've always wanted this and could not realize it 100%. The reason is the following: when you
> use the event handler of FG, you have to write functions that catch the event (for instance "mouse
> wheel" for zoom), do whatever processing and then if you end the function with:
>
> 1) return, 0: then IDL won't process the specific event at all (no zooming will be performed)
>
> 2) return, 1: then IDL will process the event normally (zooming will be performed)
>
> So now you have two ways out of this:
>
> a) make use of option 1) from above and process the event yourself. This means zoom the
> image yourself.
>
> b) make use of option 2) and with a delay (timer) do some post-event processing.
>
> I tried to do this for the zoom for a completely different problem (moving annotation on the
> image with the image when zooming and panning) and I found in the course of this another
> problem that I reported here: http://idl.marchetto.de/annotations-on-images/
> Nevertheless you might still find it useful. For example, one of the most important things you
> need to know to handle the zoom is the zooming factor. In FG this is given by:
> zoomFactor = (Delta gt 0) ? 1.25d : 1/1.25d
> where Delta is given in the call to the function:
> function WheelEvent, oWin, xPos, yPos, Delta, KeyMods
>
> In case you're going for the pan option too, what you could do, is process the mouse up option.
> First make notice if you're dealing with pan (mouse pressed + move), then in the mouse up
> function, get the new axes values and send them over to the other plot.
>
> Hope it helps and if you find a better solution, plz let me know.
>
> Regards,
> Helder
```

Dear Helder

I would like to write you because I need help about IDL program. Can you send me your email please(my email braveheart197315@yahoo.com)

Best Regard

N.S.David

Subject: Re: IDL equivalent to MATLAB's linkaxes

Posted by [Helder Marchetto](#) on Wed, 14 Oct 2015 09:43:42 GMT

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

On Wednesday, October 14, 2015 at 11:12:32 AM UTC+2, bravehea...@yahoo.com wrote:

> On Friday, 2 October 2015 21:32:08 UTC+1, Helder wrote:

>> On Friday, October 2, 2015 at 9:48:57 PM UTC+2, Gordon Farquharson wrote:

>>> Hi All

>>>

>>> Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something one would have to implement by updating each figure's xrange and yrange properties when some zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image function [2] be the best way to go?

>>>

>>> Gordon

>>>

>>> [1] <http://www.mathworks.com/help/matlab/ref/linkaxes.html?refres h=true>

>>> [2] <http://www.exelisvis.com/docs/graphicseventhandler.html>

>>

>> Hi,

>> I've always wanted this and could not realize it 100%. The reason is the following: when you use the event handler of FG, you have to write functions that catch the event (for instance "mouse wheel" for zoom), do whatever processing and then if you end the function with:

>>

>> 1) return, 0: then IDL won't process the specific event at all (no zooming will be performed)

>>

>> 2) return, 1: then IDL will process the event normally (zooming will be performed)

>>

>> So now you have two ways out of this:

>>

>> a) make use of option 1) from above and process the event yourself. This means zoom the image yourself.

>>

>> b) make use of option 2) and with a delay (timer) do some post-event processing.

>>

>> I tried to do this for the zoom for a completely different problem (moving annotation on the image with the image when zooming and panning) and I found in the course of this another problem that I reported here: <http://idl.marchetto.de/annotations-on-images/>

>> Nevertheless you might still find it useful. For example, one of the most important things you need to know to handle the zoom is the zooming factor. In FG this is given by:

>> zoomFactor = (Delta gt 0) ? 1.25d : 1/1.25d

>> where Delta is given in the call to the function:

>> function WheelEvent, oWin, xPos, yPos, Delta, KeyMods

>>

>> In case you're going for the pan option too, what you could do, is process the mouse up option. First make notice if you're dealing with pan (mouse pressed + move), then in the mouse up function, get the new axes values and send them over to the other plot.

>>

>> Hope it helps and if you find a better solution, plz let me know.

>>

>> Regards,

>> Helder

>
> Dear Helder
> I would like to write you because I need help about IDL program. Can you send me your email
please(my email xxx)
> Best Regard
> N.S.David

Hi,
maybe you should post your questions here...
Cheers,
Helder

Subject: Re: IDL equivalent to MATLAB's linkaxes
Posted by [braveheart197315](#) on Wed, 21 Oct 2015 13:30:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, 14 October 2015 10:43:45 UTC+1, Helder wrote:
> On Wednesday, October 14, 2015 at 11:12:32 AM UTC+2, bravehea...@yahoo.com wrote:
>> On Friday, 2 October 2015 21:32:08 UTC+1, Helder wrote:
>>> On Friday, October 2, 2015 at 9:48:57 PM UTC+2, Gordon Farquharson wrote:
>>>> Hi All
>>>>
>>>> Is there a canned IDL equivalent to MATLAB's linkaxes command [1], or is this something
one would have to implement by updating each figure's xrange and yrange properties when some
zoom event occurs? If the latter, would using the EVENT_HANDLER property for the image
function [2] be the best way to go?
>>>>
>>>> Gordon
>>>>
>>>> [1] <http://www.mathworks.com/help/matlab/ref/linkaxes.html?refres h=true>
>>>> [2] <http://www.exelisvis.com/docs/graphicseventhandler.html>
>>>
>>> Hi,
>>> I've always wanted this and could not realize it 100%. The reason is the following: when you
use the event handler of FG, you have to write functions that catch the event (for instance "mouse
wheel" for zoom), do whatever processing and then if you end the function with:
>>>
>>> 1) return, 0: then IDL won't process the specific event at all (no zooming will be performed)
>>>
>>> 2) return, 1: then IDL will process the event normally (zooming will be performed)
>>>
>>> So now you have two ways out of this:
>>>
>>> a) make use of option 1) from above and process the event yourself. This means zoom the
image yourself.
>>>
>>> b) make use of option 2) and with a delay (timer) do some post-event processing.

>>>
>>> I tried to do this for the zoom for a completely different problem (moving annotation on the image with the image when zooming and panning) and I found in the course of this another problem that I reported here: <http://idl.marchetto.de/annotations-on-images/>
>>> Nevertheless you might still find it useful. For example, one of the most important things you need to know to handle the zoom is the zooming factor. In FG this is given by:
>>> zoomFactor = (Delta gt 0) ? 1.25d : 1/1.25d
>>> where Delta is given in the call to the function:
>>> function WheelEvent, oWin, xPos, yPos, Delta, KeyMods
>>>
>>> In case you're going for the pan option too, what you could do, is process the mouse up option. First make notice if you're dealing with pan (mouse pressed + move), then in the mouse up function, get the new axes values and send them over to the other plot.
>>>
>>> Hope it helps and if you find a better solution, plz let me know.
>>>
>>> Regards,
>>> Helder
>>
>> Dear Helder
>> I would like to write you because I need help about IDL program. Can you send me your email please(my email xxx)
>> Best Regard
>> N.S.David
>
> Hi,
> maybe you should post your questions here...
> Cheers,
> Helder
Dear Helder,
Thanks for you response, I have two images in one window how can read these images by cursor without appear black window(because now when I do run will appear black window and I don't need this window). what is the best command used to solve this problem.
thank for your help
N.S.David
PhD Sudent-medical physics
