Subject: Interesting results when using IMAGE function
Posted by Steve Super on Fri, 15 Jan 2016 22:11:09 GMT
View Forum Message <> Reply to Message

I apologize ahead of time for the long post, but I think some background helps give some perspective. I am attempting to plot some lidar data using the IMAGE function and am running into some problems. Initially I tried to use the X and Y arguments, however this failed during the automatic gridding process. I then sought another efficient way to approach this problem and get the desired results.

I have a data array of dimensions 1830x545 along with an X array with 1830 elements and Y array with 545 elements, which represent ground location and altitude, respectively. My way around using the auto gridding was to create 2D "images" out of the X and Y arrays using REBIN. This gives me 3 arrays of the same size which I then try to display using the IMAGE function with the 'image_dimensions' and 'image_location' arguments to plot all three, in hopes of hiding the X and Y images, while still using them to create the correct axes for my data.

So my point in all of this is after displaying the X and Y arrays, the range of values is no longer the same as the max/min of the initial array. For example, when working with the altitude array, the data range gets messed up.

First I display the image and draw the y-axis the using:

```
  alt = IMAGE(vfm_alt, image_dim=[1830,545], image_loc=low_left, axis_style=4)
  y_ax = AXIS('Y', target=alt, location='left')
```

At this point I noticed the range of the axis was still in device(pixel) coordinates instead of the actual altitude values. So I type in the following commands to double check:

```
  print, alt.min,  alt.max
```

which prints the (correct) values:

```
  -0.45618850
   29.975952
```

However when I type the following:

```
  print, alt.yrange
```

I get '-0.45618850   544.54381'

So for some reason it is using the correct lower bound for the data range, but incorrectly using the array dimensions (or close to it) for the upper bound. Anyone have an idea as to what could be going on here?

-Steve

P.S. I understand this may not be the best way to do this, however after trying a few other things and not succeeding with them either I decided to think outside the box a bit.

Subject: Re: Interesting results when using IMAGE function
Posted by Paul Van Delst[1] on Tue, 19 Jan 2016 20:12:47 GMT
View Forum Message <> Reply to Message

Hello,

Maybe judicious usage of a x/y/zstyle keyword in the AXIS command?

I've had sort-of similar issues, but with regular plots.

(e.g.
https://groups.google.com/d/msg/comp.lang.idl-pvwave/G7RezL0 Y8WQ/ourv1kwuHOoJ)

cheers,

paulv

On 01/15/16 17:11, Steve Super wrote:
> I apologize ahead of time for the long post, but I think some
> background helps give some perspective. I am attempting to plot some
> lidar data using the IMAGE function and am running into some
> problems. Initially I tried to use the X and Y arguments, however
> this failed during the automatic gridding process. I then sought
> another efficient way to approach this problem and get the desired
> results.
>
> I have a data array of dimensions 1830x545 along with an X array with
> 1830 elements and Y array with 545 elements, which represent ground
> location and altitude, respectively. My way around using the auto
> gridding was to create 2D "images" out of the X and Y arrays using
> REBIN. This gives me 3 arrays of the same size which I then try to
> display using the IMAGE function with the 'image_dimensions' and
> 'image_location' arguments to plot all three, in hopes of hiding the
> X and Y images, while still using them to create the correct axes for
> my data.
>
> So my point in all of this is after displaying the X and Y arrays,
> the range of values is no longer the same as the max/min of the
> initial array. For example, when working with the altitude array, the
> data range gets messed up.
>
> First I display the image and draw the y-axis the using: alt =
> IMAGE(vfm_alt, image_dim=[1830,545], image_loc=low_left,
> axis_style=4) y_ax = AXIS('Y', target=alt, location='left')
>
> At this point I noticed the range of the axis was still in
> device(pixel) coordinates instead of the actual altitude values. So I
> type in the following commands to double check:
>

> print, alt.min, alt.max which prints the (correct) values:
> -0.45618850 29.975952
>
> However when I type the following: print, alt.yrange I get
> '-0.45618850 544.54381'
>
> So for some reason it is using the correct lower bound for the data
> range, but incorrectly using the array dimensions (or close to it)
> for the upper bound. Anyone have an idea as to what could be going on
> here?
>
> -Steve
>
> P.S. I understand this may not be the best way to do this, however
> after trying a few other things and not succeeding with them either I
> decided to think outside the box a bit.
>

Subject: Re: Interesting results when using IMAGE function
Posted by Matt Haffner on Thu, 21 Jan 2016 20:36:01 GMT
View Forum Message <> Reply to Message

Steve,

You need to give IMAGE the values in your data units for X and Y somehow. Have you tried just:

```
alt = image(vfm_alt, x, y)
```

IDL should use the range of values in X for the x-axis and Y for the y-axis--you mention that they are already the proper lengths for the image you are displaying.

In the example you give, IMAGE_DIM is supposed to be in your data units, not pixel units. So, the reason alt.yrange is returning ~ 0 - 545 is because that's what you've told it the Y-axis range is with IMAGE_DIM.

The alt.min and alt.max are returning the range of vfm_alt--the data your alt IMAGE is displaying.

An aide: if you already have an original data array that's 1830x545, IDL shouldn't do any auto-gridding in a call to IMAGE. Is it perhaps a scaling issue instead?

Hope that helps a bit...

- mh

On Friday, January 15, 2016 at 4:11:15 PM UTC-6, Steve Super wrote:
> I apologize ahead of time for the long post, but I think some background helps give some perspective. I am attempting to plot some lidar data using the IMAGE function and am running into

some problems. Initially I tried to use the X and Y arguments, however this failed during the automatic gridding process. I then sought another efficient way to approach this problem and get the desired results.
>
> I have a data array of dimensions 1830x545 along with an X array with 1830 elements and Y array with 545 elements, which represent ground location and altitude, respectively. My way around using the auto gridding was to create 2D "images" out of the X and Y arrays using REBIN. This gives me 3 arrays of the same size which I then try to display using the IMAGE function with the 'image_dimensions' and 'image_location' arguments to plot all three, in hopes of hiding the X and Y images, while still using them to create the correct axes for my data.
>
> So my point in all of this is after displaying the X and Y arrays, the range of values is no longer the same as the max/min of the initial array. For example, when working with the altitude array, the data range gets messed up.
>
> First I display the image and draw the y-axis the using:
> alt = IMAGE(vfm_alt, image_dim=[1830,545], image_loc=low_left, axis_style=4)
> y_ax = AXIS('Y', target=alt, location='left')
>
> At this point I noticed the range of the axis was still in device(pixel) coordinates instead of the actual altitude values. So I type in the following commands to double check:
>
> print, alt.min, alt.max
> which prints the (correct) values:
> -0.45618850
> 29.975952
>
> However when I type the following:
> print, alt.yrange
> I get '-0.45618850 544.54381'
>
> So for some reason it is using the correct lower bound for the data range, but incorrectly using the array dimensions (or close to it) for the upper bound. Anyone have an idea as to what could be going on here?
>
> -Steve
>
> P.S. I understand this may not be the best way to do this, however after trying a few other things and not succeeding with them either I decided to think outside the box a bit.
